RAPPORT ANNEE 2020 – 2021

TITRE:

Development of a methodology for calibration and quantification of uncertainties of hydraulic models using meta-model.

Sous-titre (éventuel) : /

_

Nom de l'entreprise : Compagnie Nationale du Rhône (CNR)

Adresse : 2 Rue André Bonin, 69004 Lyon

Nom et prénom du Maître de Stage (dans l'entreprise / le laboratoire) :

Mr. DURON Luc

Nom et prénom du Tuteur Ecole

Pr DUFOUR Frédéric

Stage de :

☐ **1A Découverte de l'Entreprise** **Série** : □ A1 ☐ A2 ☐A3 ☐B1 ☐B2 ☐B3 ☐C1 ☐C2 ☐C3

☐ **2A Assistant Ingenieur** **Filière :** ☐ASI☐HOE☐GEN ☐IEE☐ME ☐SEM ☐ADMIS/T ☐MASTER ☐ETUDIANTS A L'ETRANGER ☐AUTRES

☑ **3A Projet de Fin d'Etudes** **Filière :** ☐ASI☐HOE☐GEN ☐IEE☐ME ☐SEM ☑MASTER ☐ETUDIANTS A L'ETRANGER ☐AUTRES

Nom et prénom de l'étudiant : ALLOUL Badre Abderrahmane

Sujet confidentiel : Oui ☐

Un rapport confidentiel ne pourra être communiqué à quiconque (étudiant, enseignant, personne extérieure) sauf autorisation spéciale accordée par le Maître de Stage dans l'entreprise/le laboratoire

Acknowledgments

I would like to thank all the people who contributed to the success of my internship and who helped me in writing this report.

First of all, I address my thanks to my professor, **Pr. EMERIAULT Fabrice**, who helped me a lot in my search for an internship and allowed me to apply to this company. His advice allowed me to target my applications and went for this internship which was in total adequacy with my expectations.

A special thanks to my school advisor and Professor, **Pr. DUFOUR Frédéric** who encouraged me a lot during the internship and valorized my work.

I would like to warmly thank my internship supervisor, **Mr. DURON Luc**, for his welcome, the time spent together and the sharing of his expertise on a daily basis He taught me and helped me a lot in this journey and introduced me to the field on which I will build my academic career. Also, thanks to his confidence I was able to fulfill myself completely in my missions. He was of great help in the most delicate moments.

I also thank the whole CNR numerical modeling team for their welcome, their team spirit, and in particular **Mr. BENEFICE Guillaume**, **Mr. CIERCO François-Xavier** who helped me a lot to understand the issues and to validate the different internship results levels.

Finally, I would like to thank all the people who advised and reread me during the drafting of this internship report: my family, my friend **ACHOURI Redouane**.

Abstract

Developing precise numerical models that can reproduce reality, has been through a lot of development and improvement during the last decades in parallel with the rise of many technologies and programming libraries that have enlarged many methods' application potential. Therefore, a contribution to this part of the numerical modeling world seemed to be valuable, and building a fast-hydraulic numerical model from which engineers could derive conclusions and field decisions, was the main motivation for this work.

In a 1D hydraulic model, calibration is the core step, thus, the first part of this work focused on its improvement and modernization. It starts by defining the calibration procedure and methods in a classic 1D hydraulic modeling framework by specifying the hydraulic parameters that can be involved in this step that they are: Manning-Strickler and discharge coefficient. Starting with a single calibration parameter case which is the Strickler coefficient, it introduces a new way of calibrating hydraulic models using a surrogate model which is less time-consuming, then how it can be coupled with a hydraulic simulation code to form what is called a grey-box which is a machine learning framework for models calibration. Each tool's mathematics used behind, different algorithms, concepts, and optimization algorithms were explained thoroughly. To be able to perform a fair comparison between those pure mathematical techniques, this work provided two new generic metrics that will allow engineers to assess, compare and get insights about their calibration tools. A case study with real field data was then chosen to observe those tool's performance and capabilities. After using a single parameter, the second one was involved, and another case study was chosen to analyze the metamodel response which did not provide a satisfying prediction and dispersion results.

The second part of this work evokes the methodology and the workflow to conduct a sensitivity analysis and uncertainty propagation analysis in the hydraulic model. Firstly, it describes the generic methodology to perform a global sensitivity analysis (GSA hereinafter). It starts by presenting the core objective behind a GSA study and its different steps from the sampling method to the Sobol indices computation. That was possible through a case study where the uncertain parameters and the model response variable of interest were pointed out and each step was explained and interpreted in detail. As the first part of this work confirmed that metamodel is so fast and are prioritized for time-consuming analysis, the Gaussian process was involved as a metamodel to replace the classical hydraulic model and it allows us to conduct another GSA study with identical parameters to be able to compare the final results and again metamodel was negligible in time consumption and provide similar results that turn out to similar conclusion on impactful uncertain parameters in our case which relies mostly on Manning-Strickler coefficient.

As an overall takeaway of this work, metamodels are the adaptive way to perform huge time-consuming for physical modeling projects under actual hardware resources constraints, and as it is all-based on efficient model description including the parameters, the input-output relationship, the GSA is a mandatory step to maximize the metamodel performance and to reduce prediction error.

Keywords: Hydraulic modeling, calibration, Strickler coefficient, simulation, Metamodel, Machine learning, Gaussian process, Kernel, linear regression, XGBoost, Scikit-learn, Optimization, OpenTurns, sensitivity analysis, uncertainty propagation, Probability distribution, Sobol indices

Résumé

Le développement de modèles numériques précis capables de reproduire la réalité a connu de nombreux développements et améliorations au cours des dernières décennies, parallèlement à l'essor de nombreuses technologies et bibliothèques de programmation qui ont élargi le potentiel d'application de nombreuses méthodes. Par conséquent, une contribution au développement de la modélisation numérique semblait être précieuse, et la construction d'un modèle numérique hydraulique rapide à partir duquel les ingénieurs pourraient tirer des conclusions et des décisions sur le terrain, était la principale motivation de ce travail.

Dans un modèle hydraulique, la calibration est l'étape centrale, ainsi, la première partie de ce travail s'est concentrée sur son amélioration et sa modernisation. Ce travail commence par définir la procédure de calibration dans un cadre classique de modélisation hydraulique 1D en précisant les paramètres hydrauliques pouvant intervenir dans cette étape qu'ils soient : coefficient de frottement (coefficient de Strickler) et coefficient de débit. Partant d'un seul cas de paramètre d'étalonnage qui est le coefficient de Strickler, ce travail introduit une nouvelle façon d'étalonner des modèles hydrauliques à l'aide d'un modèle de substitution ou ce qu'on appelle un métamodèle qui prend moins de temps, puis comment il peut être couplé à un code de simulation hydraulique pour former ce qu'on appelle un *grey-box* qui est un *framework* d'apprentissage automatique pour l'étalonnage de modèles. Les mathématiques de chaque outil utilisé et les différents concepts et algorithmes d'optimisation ont été expliqués en détails. Pour pouvoir effectuer une comparaison équitable entre ces techniques mathématiques pures, ce travail a fourni deux nouvelles métriques génériques qui permettent aux ingénieurs d'évaluer et de comparer la performance de leurs outils d'étalonnage. Une étude de cas réel a ensuite été choisie pour analyser les capacités de chaque outil. Après avoir utilisé un seul paramètre, la seconde étape a été impliqué et une autre étude de cas a été choisie pour analyser une autre fois la réponse du métamodèle. Dans ce cas les résultats de prédiction et de dispersion n'étaient pas satisfaisants ce qui fera un sujet prochain pour des améliorer ces résultats.

La deuxième partie de ce travail évoque la méthodologie et le *workflow* pour réaliser une analyse de sensibilité et une analyse de propagation des incertitudes dans le modèle hydraulique. Dans un premier temps, il décrit la méthodologie générique pour effectuer une analyse de sensibilité globale (ci-après GSA pour *Global sensitivity analysis*). Il commence par présenter l'objectif principal d'une étude GSA et ses différentes étapes, de la méthode d'échantillonnage au calcul des indices de Sobol. Cela a été possible grâce à une étude de cas où les paramètres incertains et la variable de d'intérêt ont été choisie et chaque étape a été expliquée et interprétée en détails. Comme la première partie de ce travail a confirmé que le métamodèle est si rapide et prioritaire pour les analyse lourdes en temps de calcul, le processus gaussien a été impliqué en tant que métamodèle pour remplacer le modèle hydraulique classique et il nous permet de mener une autre étude GSA avec des paramètres identiques ce qui justifie la comparaison des résultats finaux et encore une fois le métamodèle était négligeable en temps de calcul et il a fourni des résultats similaires qui aboutissent à une conclusion similaire sur les paramètres incertains ayant un impact dans notre cas, qui repose principalement sur le coefficient de Strickler.

En tant que conclusion globale de ce travail, les métamodèles sont le moyen adaptatif pour effectuer une calibration des modèles hydrauliques qui prend beaucoup de temps sous la contrainte de ressources matérielles réelles, et comme la technique des métamodèles est entièrement basé sur une description du modèle physique efficace, y compris les paramètres, les relation entrées-sorties,

rends la GSA une étape obligatoire pour maximiser leurs performances et réduire les erreurs de prédiction.

Mots clés : Modélisation hydraulique, calage, coefficient de Strickler, simulation, Métamodèle, Machine Learning, Krigeage, régression linéaire, XGBoost, Scikit-learn, Optimisation, OpenTurns, analyse de sensibilité, propagation de l'incertitude, Indice de Sobol

Table of contents

Liste of figures :

Liste of figures :

List of appendices :

1 Introduction

Water level modeling is a critical step in flood warning systems. A high-quality forecast requires the development of a hydraulic model that reliability accounts for the main sources of uncertainty.

1-D open channel hydraulic models are used for a wide range of field application purposes including extreme event discharge prediction and flood impact area extension. Limitation can be experienced e.g., for slightly overbank flows in compound channels, for important spatial heterogeneity, roughness, *etc.* Such limitations can be explained theoretically by various arguments [1]. Among those reasons, some are related to the use of the Manning equation (Laushey, 1989, Pappenberger et al., 2005) and the lack of proper modeling of the different energy loss terms [2].

Most open-channel hydraulic models are based on Saint-Venant equations. They estimate water level and flow velocity through time-limited computations. The equations system is generally closed with the Manning-Strickler equation, which includes Strickler or Manning coefficients. These coefficients account for dissipation phenomena. They depend on physical riverbed characteristics, such as the roughness of the bottom of the river, the presence of plants, but also geometric ones, such as the shape of the riverbed, and meanders. Because the dissipation terms of the Navier-Stokes equations mainly involve interactions between the different velocity components, they cannot be explicitly represented in 1D models.

Therefore, in 1D models, most dissipation phenomena are roughly modeled through the Manning-Strickler equation which provides a global relationship between the flow rate and the head losses. As a result, the number of dissipation terms implicitly represented by the Strickler coefficients either depends on the problem complexity or the discharge.

2 Compagnie Nationale du Rhône (CNR) -- The hosting company

CNR is the concessionary of the River Rhone for three-fold purposes: hydroelectricity production, navigation, and irrigation. It is a French leading producer of exclusively renewable energy. Its expertise in the entire value chain of hydro-, solar and wind power and its missions in the general interest make it a major partner in the development and balance of the territories. Its project efficiency and rentability are acknowledged internationally.

CNR has designed and built **19 hydropower plants**, **19 dams,** and **14 locks**, opened **330 km of navigable waterway** between Lyon and the Mediterranean Sea, and developed industrial and port sites (See their localizations on Appendix 1). Solidarity lies at the heart of CNR's actions and it is actively involved in the development of renewable energies (**hydroelectricity**, **solar energy,** and **wind power**), in support of territorial development and energy transition.

2.1 *D*irection d'*I*ngénierie des *G*rands *P*rojets (DIGP) -- The hosting department

DIGP department is the integrated engineering department of CNR with a wide array of competencies and it has more than a hundred engineers and technicians. My internship was performed in the hydraulic and hydrology team (17 collaborators).

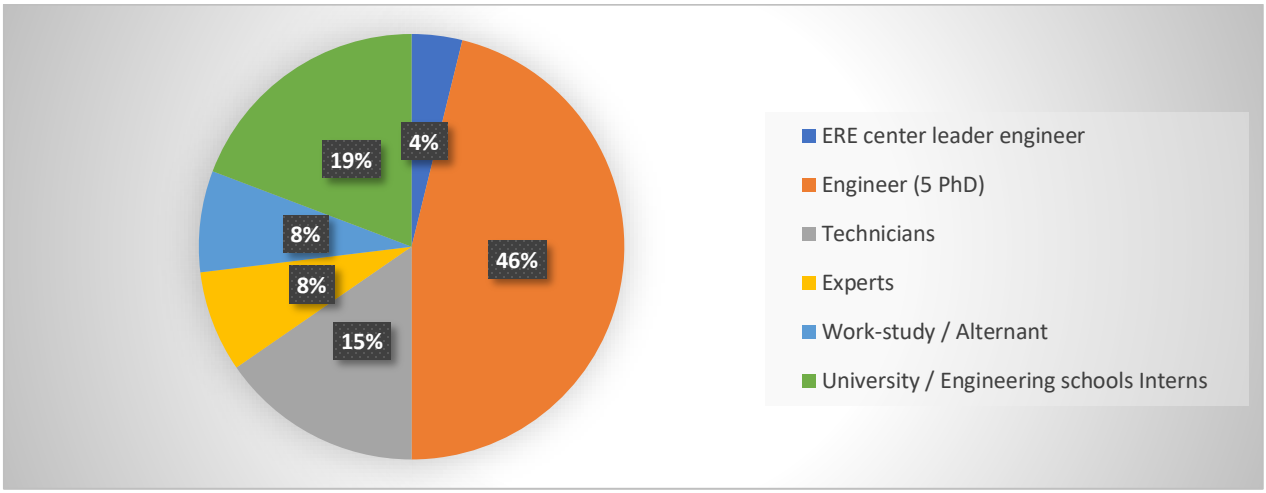


Figure 1: DIGP department collaborators repartition

2.1.1 Missions

DIGP's main missions are:

1. Maintain and develop the technical expertise essential to the CNR concessionary's mission.
2. Strengthen the mastery of the project cycle.
3. Promote the know-how of service provision in France and internationally, by developing its portfolio of projects.

DIGP department gathers multiple engineering centers:

- ERE : "Ecoulement et ressource en eau"
- ELM: "Electromécanique"
- ENV: "Environnement"
- GCR: "Genie civil"
- TMT : "Travaux méthodes et transfert"

ERE center, the hosting center assembles project managers among 5 hydrologists and 12 river hydraulic engineers. They are responsible for hydraulic projects including probabilistic event forecasting, innovative tools, and methods development, and water resource optimization. Technically, an engineer at the ERE center of DIGP works daily on many 1-2-3 D modeling projects and performs potential risk studies as flood propagation area studies. Moreover, they design hydro-electric power stations and ensure ecological stability over the Rhône River by conducting environmental studies to preserve the ecological diversity and quantifying the impact of engineering projects on nature and mainly on the water behaviors in river channels and floodplains before and after hydraulic events.

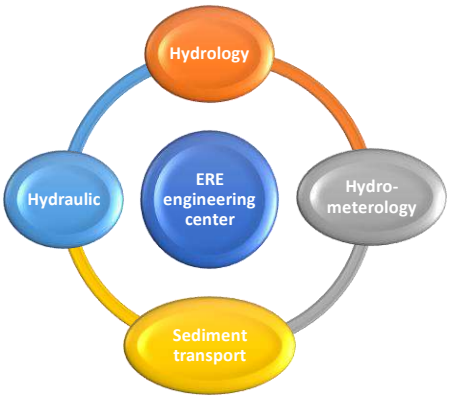


Figure 2:ERE engineering center principal expertise field

3 What is a hydraulic model?

Modeling is a mathematical or physical description of a physical process either using mathematical language or general logical frameworks.

The academic definition of a **hydraulic model** is a mathematical model of a fluid flow system. These models can inform decisions about selecting and implementing flood reduction and/or restoration projects. It also satisfies regulatory requirements and ensures that natural, ecological resources, and urban systems are not damaged by flooding induced by modifications to creeks, rivers, and channels made for human exploitation purposes [3].

3.1.1 CNR's Hydraulic numerical code – Crue10

Through the last two decades, CNR has been developing a numerical solution code of the shallow water equation coupled with a pre-and post-processing tool.

Crue10 numerical code needs multiples inputs file to run and provide outputs variables of interest. The Crue10 GUI Fudaa-Crue has simplified the generation process for further external studies. It means, that you can create your hydraulic model in Fudaa-Crue then apply external scripts or specific tasks by manipulating the Fudaa-Crue's generated files.

The Fudaa-Crue model building resulting files are *.xml type and four of them will be extensively used to conduct external (apply external algorithms on those files) different studies in this report:

- *.etu.xml: Steering file containing other involved files paths.
- *.aoc.xml: Is the paramount file for the calibration process as it contains all model water profile, their ponderations coefficients $\boldsymbol{\alpha}$ [1], all the Strickler coefficients by zones and their ranges values, the discharge coefficient interval limit ΔQ (ΔQ at a specific model node).
- *.lhpt.xml: It contains all the model cross-sections with water levels in each cross-section depending on the considered water profile
- *.dclm.xml: It contains the flow condition (steady or unsteady flow) and model nodes values for each water profile

Examples of these xml files are available in Appendix 2, Appendix 3, Appendix 4, and Appendix 5

[1] The ponderation coefficient can be attributed to a specific water level profile and it represents the contribution percentage of the water profile in the cost function in the optimization step.

3.1.2 Telemac-2D

It is a code that solves depth-average free surface flow equations as derived first by Barré de Saint-Venant. The results at each node of the computational mesh are the depth of water and the depth-average velocity components. The principal application of the TELEMAC-2D is in free-surface maritime or river hydraulics.

This code is based on high-capacity algorithms developed according to the finite-element method. To lighten the computations and optimize simulation time, the code is launched by command lines, nevertheless the post- and pre-processing can be performed on a user-friendly interface with a bunch of extremely sophisticated functions.

TELEMAC-MASCARET system consists of multiple modules and the most important modules from a frequency use point of view are:

- TELEMAC-2D: Two-dimensional flows - Saint-Venant equations (including transport of a diluted tracer)
- SISYPHE 2D: sediment transport computation module. It can be used to model complex morphodynamics processes in diverse environments, such as coastal, rivers, lakes, and estuaries, for different flow rates, sediment size classes, and sediment transport modes. [4]
- GAIA: This is the new open-source TELEMAC module for sediment transport and bed evolution based on the well-known and widely used sediment transport module SISYPHE.

4 Hydraulic model calibration

In river hydraulic, the main calibration objective is to adapt the roughness coefficients over the river channel and floodplains, or even the design of the structures (mainly culverts and bridges), to reproduce the water surface profiles (**Figure 3)** observed (hydrometric stations historical records during past floods events) or simulated (numerical code simulation results).

For our study cases and after being proposed by [5], model calibration and validation based on water level and flow observations became a necessary step to interrogate hydraulic model capability to provide reality-like results. The goal of the calibration process of a 1D hydraulic model is to find and introduce into the model, the roughness coefficients (Strickler) which make it possible to best represent the hydraulic events observed previously to be able to extrapolate on future extreme events. Most often, available data on the real events are water surface profiles (Water levels at stations along the river. **Figure 3**), associated with boundary conditions.

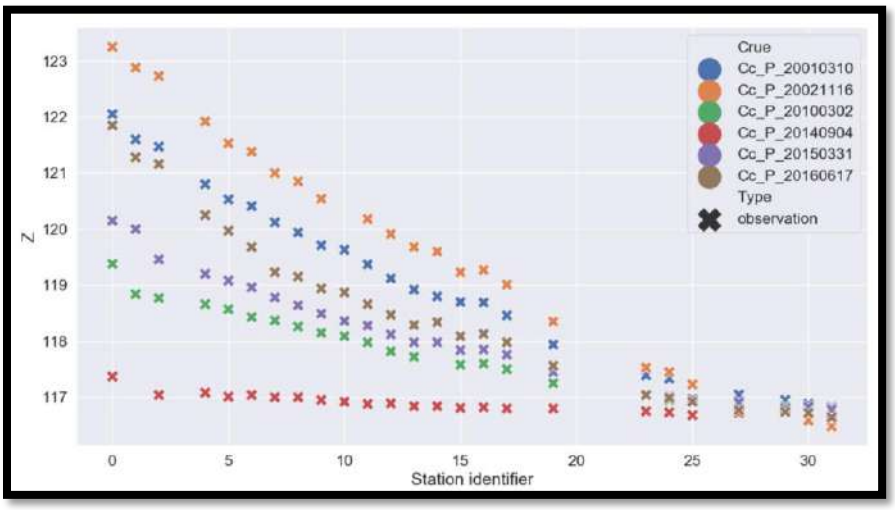


Figure 3: Water level observed at the different stations (6 profiles at 32 sections)

This is a mathematical simple description of the calibration process. Let f denote the hydraulic model. For a given boundary condition q, we have

$$f_q : \mathcal{D} \subset \mathbb{R}^d \to \mathbb{R}^{N_s}, \quad k = \begin{pmatrix} k_1 \\ \vdots \\ k_d \end{pmatrix} \mapsto f_q(k) = \begin{pmatrix} f_q^1(k) \\ \vdots \\ f_q^{N_s}(k) \end{pmatrix}$$

with k a vector of friction coefficients (Strickler) in d zones, and $f_q(k)$ a vector of water levels at N_s sections. One zone contains several contiguous sections (N_s>d).

Calibrating the hydraulic model means minimizing the errors between hydraulic model results and observations.

$$k_{opt} \in \underset{k \in \mathcal{D}}{\operatorname{argmin}} J(k) \qquad \text{with} \begin{cases} J(k) = \dfrac{1}{N_c} \displaystyle\sum_{i=1}^{N_c} J_{q_i}(k)\, \alpha_i \\ J_{q_i}(k) = \sqrt{\dfrac{1}{N_s} \displaystyle\sum_{s=1}^{N_s} \left(f_{q_i}^s(k) - O_i^s\right)^2} \qquad \forall i \in \{1, \dots, N_c\} \end{cases}$$

With N_s is the number of sections, N_c is the number of water level profiles, k is a vector of friction coefficients, $f_{q_i}^s(k)$ is the hydraulic model water level prediction computed with the flow rate q_i and section s, O_i^s is the water level observation measured for the flow rate q_i and section s α_i is the ponderation of the ith water level profile.

Determining an optimal vector k_{opt} with an optimization algorithm would imply repeated calls to the hydraulic model. Since evaluating the hydraulic model is a time-consuming process, we use a simpler metamodel $\hat{f}$ to replace it. The computational cost of the optimization step is therefore reduced, and the problem becomes:

$$k_{opt} \in \underset{k \in \mathcal{D}}{\operatorname{argmin}}\, \hat{J}(k) \qquad \text{with} \begin{cases} \hat{J}(k) = \dfrac{1}{N_c}\displaystyle\sum_{i=1}^{N_c} \hat{J}_{qi}(k)\alpha_i \\ \hat{J}_{qi}(k) = \sqrt{\dfrac{1}{N_s}\displaystyle\sum_{i=1}^{s}\left(\hat{f}_{q_i}^{s}(k) - O_i^s\right)^2} \quad \forall i \in \{1, \dots, N_c\} \end{cases}$$

with k a vector of friction coefficients (Strickler) in d zones, and $f_q(k)$ a vector of water levels at N_s sections. One zone gathers several contiguous sections (N_s>d).

Calibrating the hydraulic model means minimizing the errors between hydraulic model results and observations.

In this work, automated Strickler coefficient calibration is performed under a steady flow assumption. For steady flows, the calibration is defined as an optimization problem on the root-mean-square error ($\hat{\boldsymbol{J}}(\boldsymbol{k})$ or RMSE$_{(i)}$) computed with simulated ($\hat{\boldsymbol{f}}_{\boldsymbol{q_i}}^{\boldsymbol{s}}(\boldsymbol{k})$) and measured water levels ($\boldsymbol{O_i^s}$) along with the domain as the objective function. The calibration step consists in finding a set of Strickler coefficients (k_{opt}) to minimize this objective function.

Note that the minimum of the objective function over the set of coefficients is not expected to be 0 (perfect calibration case) since the model is a simplification of the real behaviors of the river. Furthermore, the measured water levels are subject to the limited accuracy of the sensors. The model geometry is also affected by sediment transport so that no temporal coincidence can be observed between the bathymetry data and the measured water levels.

4.1 Calibration parameters (Features[2])

4.1.1 Strickler coefficients K

When overbank flows occur in natural rivers, they often give rise to non-uniform flows. These non-uniform flows are caused by bed level changes, longitudinal variation in channel width, and/or unbalanced discharge distribution across the channel. Currently, 1-D modeling is still commonly used for operational purposes. Some limitations are somehow experienced locally, whether the flow conditions do not fit the main assumptions of 1-D modeling (e.g., channel junctions or for slightly overbank flows). Therefore, with the advancement of flood management, investigators have proposed a framework for modeling the hydraulic and morphologic characteristics in natural waterways. The most performant approach is based on the principle of compound sections or the **DCM[3] method**. This method provides physical overall discharge and is still the primary tool used by engineers for modeling in compound channels. For example, the treatment of lateral variability in commercial software tools for one-dimensional river modeling such as SOBEK, MIKE11, and HEC-RAS is all based on DCM.

In this method, the main equation system is the Strickler equation, so that one of those coefficients accounts for all the dissipation terms, disregarding the real dominant phenomena occurring in the flow.

[2] Features are the problem characteristic that might have bearing on it and lead to the best outcomes and most beneficial insights. They must describe the inherent structure of the data.
[3] Divided channel method: Series of roughness coefficients will account for every source of energy dissipation

Strickler equation: $V = K_s \,.\, R_h^{2/3} \,.\, i^{1/2}$ [4-1]

Where:

- V is the velocity in the transversal section (m/s)
- K_s is the Strickler coefficient ($m^{1/3}/s$) ($K = \frac{1}{n}$: where n is the Manning coefficient)
- R_h is the hydraulic radius (m)
- i is the bed slope (m/m)

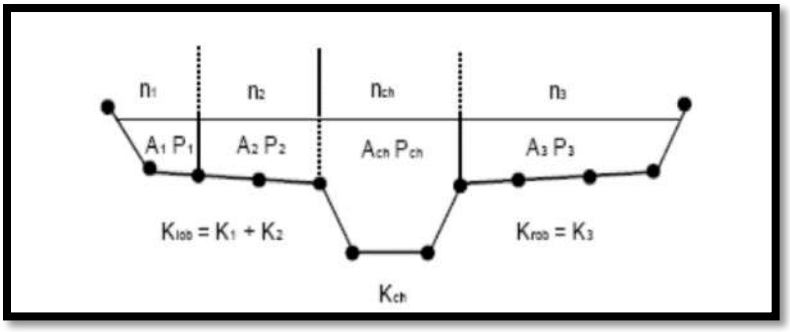


Figure 4: HEC-RAS Default Subdivision Method (Courtesy of the U.S Army Corps of Engineers)

4.1.2 Discharge ratio $Q_{factor} = \frac{Q_{simulation}}{Q_{Observation}}$

In France, the streamflow measurement network is ruled by the Ministry in charge of Environment and Infrastructures (MEDAD) through Regional Environment Agencies (DIREN) and Flood Forecast Services (SPC). CNR also produces hydrometric data and feed hydrometric databases from 150 gauging stations for the Rhône River and main tributaries. There is no universal method for measuring water streamflow. The chosen method is affected by various factors:

- Site topographical configuration and flow hydraulic conditions
- Used devices and equipment
- The fixed validity and accuracy degree.

French hydrometric services are concerned by the assessment measurements quality as the discharge is computed with rating curves (stage-streamflow relationship). The mean streamflow uncertainty was estimated to be ±10% and may be higher if it is associated with other rating curves (e.g., at hydrographic confluences). A tradeoff must be established between the density of the gauging data used to build the rating curve and the possible changes in the stage-streamflow discharge relationships over the retained period.

Therefore, it seemed interesting to integrate a new calibration correction multiplicator parameter $\boldsymbol{Q_{factor}}$ to be optimized to get the most accurate data-driven discharge prediction for further studies.

Discussion and conclusion

A question may be asked intuitively by the reader wondering about why previously mentioned parameters were chosen to be the calibration process inputs. Therefore, this question is justified, and one of the answers is that hydraulic literature, expert and some calibration sensitivity analysis studies that were conducted, showed that roughness coefficient and discharge ratio has proved to

be the main calibration parameters, nevertheless, other parameters can be considered for different event data as discharge coefficient of weirs, river geometry…

Water levels in different river sections and for multiples events, combined with calibration parameters, will be the calibration problem inputs in the next sections, regardless of used model, workflow, or case study.

4.2 Study cases

To perform the next methodology steps, we have judiciously selected three main hydraulic models from different geographical locations along the Rhône River. They present different calibration cases that a modeler can face in a usual hydraulic project. We will apply different methods, algorithms, comparison, and validation on these models to be able to conclude on model's computation performance and estimate potential further enhancements.

The three hydraulic models are extracted from the CNR intern GIS[4] : CNR-MAPS. Each model will be defined by its:

- Stickler zones and values.
- Water levels profiles are the floods events water elevations measured in different positions along with the longitudinal river channel profile or even at some floodplain points.

Strickler coefficients zones are defined as follows:

[4] Geographic information system.

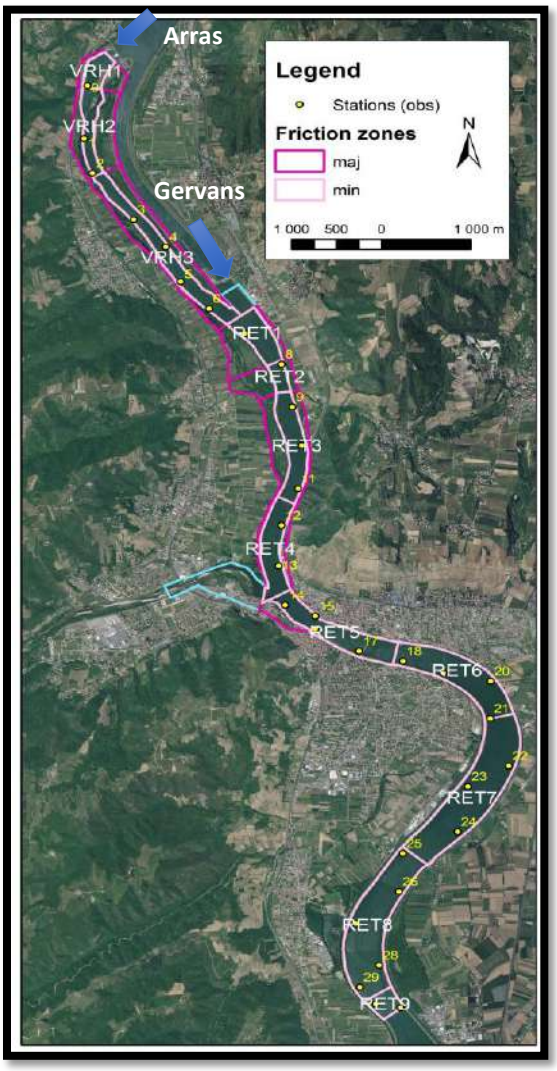


Figure 5: Illustration of the divided roughness zones in Bourg-Lès-Valence model 2016

In the illustration above the hydraulic model is separated into multiple zones (in total 19). The "Vieux-Rhône" section (before the confluence with the channel from Gervans) has 6 zones: VRH1 to VRH3, each having both minor (min) and major beds (maj). Moreover, the reservoir downstream has 13 zones (RET1 to RET9) with minor bed and major bed up to RET4 included.

In this example, the friction coefficient (Strickler law) is assumed to be in the range defined in Table 1 for different zones. The coefficient variation range is purposely wide because it is assumed to have no prior knowledge of the model.

Table 1: Friction coefficient variation range

Zones	Lower bound	Upper bound
VRH1maj to VRH3maj RET1maj to RET4maj	5	30
VRH1min to VRH3min RET1min to RET9min	20	55

Based on previous zones and Strickler ranges definitions, the models were divided exactly following the same logic in the previous example of Bourg-Lès-Valence, with different complexity levels to assist the code's capacities and highlight its limits. See Appendix 6, Appendix 7and Appendix 8 for the models' geographical views.

Table 2: Study cased configurations:

Model	Considered water level profiles	Number of observed water level	Friction zones number	The study conducted on the model
CE2016 (Chautagne)	2	15	12	• Calibration without/with Q_{factor}
BV2016 (Bourg-Lès-Valence)	5	160	19	• Calibration without Q_{factor}
CA2020 (Caderousse)	7	616	29	• Calibration without Q_{factor} • Benchmark AOC vs PYMODAST

As the physical model describing the problem is defined in detail according to the previously mentioned assumptions and considered problem parameters, the next sections will be dedicated to diving into calibration modules and how they were developed, validated, and improved to be time-efficient and provide accurate results.

4.3 AOC: The CNR's module for 1D hydraulic model calibration

To carry out a calibration process several stages are needed, and the problem is being broken down into:

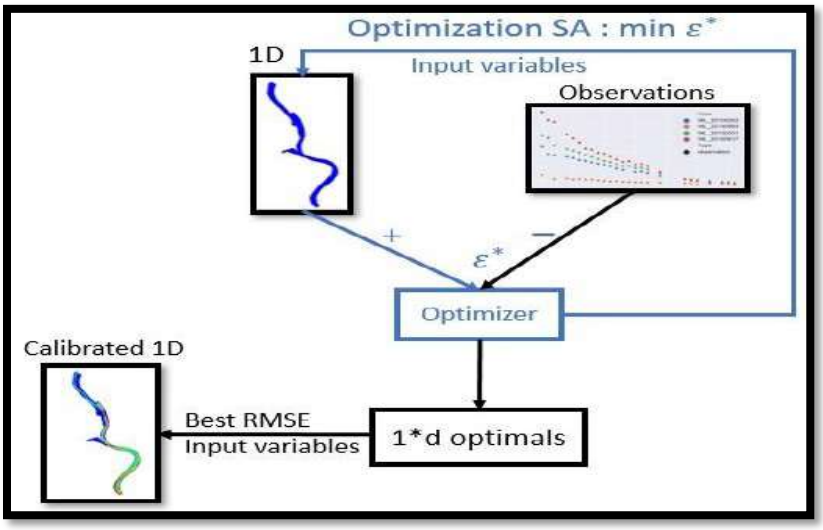
- The selection and validation of the measured data to be used.
- Calibration input model parameters and their variation ranges.
- Parameters calibration aiming to provide the best prediction results[5].

AOC is a simulated annealing (SA) algorithm (heuristic algorithm to estimate the global optimum) based module implemented in the Fudaa-Crue GUI. The best illustration of SA algorithm application is the *Travelling salesman problem* [6].
AOC module consists of the optimization of the pre-defined objective function starting from an initial state. It performs the calibration process's optimization and provides the optimal input set[6].

At each iteration, a new set of values (of the inputs variable) is generated by random perturbation of the previous state. Then, the predicted data (water levels) are computed by the numerical code which takes the newly generated set of values as its current inputs. Therefore, as the objective function is calculated at each step, if the value is lower than the previous one, the new guess is validated, and a new step begins. This method, as all the heuristics, tries to overcome the objective function local minima trap through its exploratory characteristic.
Theoretically, after several calculation steps, the algorithm converges as the changes in the guess values become infinitesimals and their differences are reduced considerably. However, a realistic

[5] Accurate prediction results values ➔ closest to the observed data
[6] Optimal input parameters combination.

practical situation consists of fixing a threshold which will be considered as a stopping criterion for the algorithm. This optimization algorithm is called multiples times (generally number of calls equal to the number of the problem parameters), then the set of parameters ensuring the minimum objective function result, will be then considered as the optimal set.



Workflow 1 : AOC tool – **A**ssistance à l'**O**ptimisation du **C**alage

In practice, the objective function can be selected according to the study aims, next section provides explanations about AOC's objective functions choice.

4.3.1 Practical description of AOC module

To calibrate the roughness coefficients, only water profile that all the differences are within the uncertainty range (+ -10cm retained and + -20cm old Rhône river stream) are considered as input data for the calibration algorithm, so that the deviations are Gaussian with zero mean and sigma standard deviation such that **4σ** which corresponds to the width of the **95%** confidence interval is less than the length of the tolerated uncertainty interval. We can summarize these considerations, by taking as a criterion the $RMSE_{min}$[7].The RMSE is implemented in AOC as the objective function to be minimized which will determine the relevance of the calibration

$$J(z_{simulation}(p), z_{obs}) = RMSE = \sqrt{\frac{\sum_{x_0}^{x_n}(z_{sim}(x_i,p) - z_{obs}(x_i))^2}{n+1}} \qquad [4\text{-}2]$$

Where: $p^* \in Arg\min J$, p is a vector of the calibration parameters of the model that minimize the objective function.

AOC was a core of many developing projects at CNR attempting to minimize computations time and large results discrepancy. This algorithm is computationally expensive since each calibration requests hundreds to a few thousands' hydraulic computations. Therefore, it could not be used for a 2D model calibration, CPU time would become prohibitive, the reasons why a new calibration process implementing a new optimization algorithm is developed to strengthen the calibration process and turn it into a usual daily working task.

The next sections include the PYMODAST calibration module description as it is built on an intermediate metamodel from hydraulic calculations and solves an optimization problem on the metamodel.

[7] Minimum root means square error.

4.4 PYMODAST: Hydraulic model semi-automatic calibration with a surrogate model approach

PYMODAST stands for Python Modeling Data Science Tools and is a tool dedicated to hydraulic model calibration with a surrogate model-based approach.

Another mathematical formulation is introduced through PYMODAST by adding water profiles weights coefficients (α_i), called ponderations as well.

A training dataset is required to build metamodel candidates based on different regression methods. It is composed of N sets of friction coefficients defined by a Design of Experiment (DOE) and water levels along with the domain. Water levels are processed by the hydraulic model for each set of friction coefficients. The metamodels accuracy is assessed via 10-fold cross-validation that computes the root-mean-square deviation (RMSD) between water levels predicted by the hydraulic model and by the metamodel along with the domain.

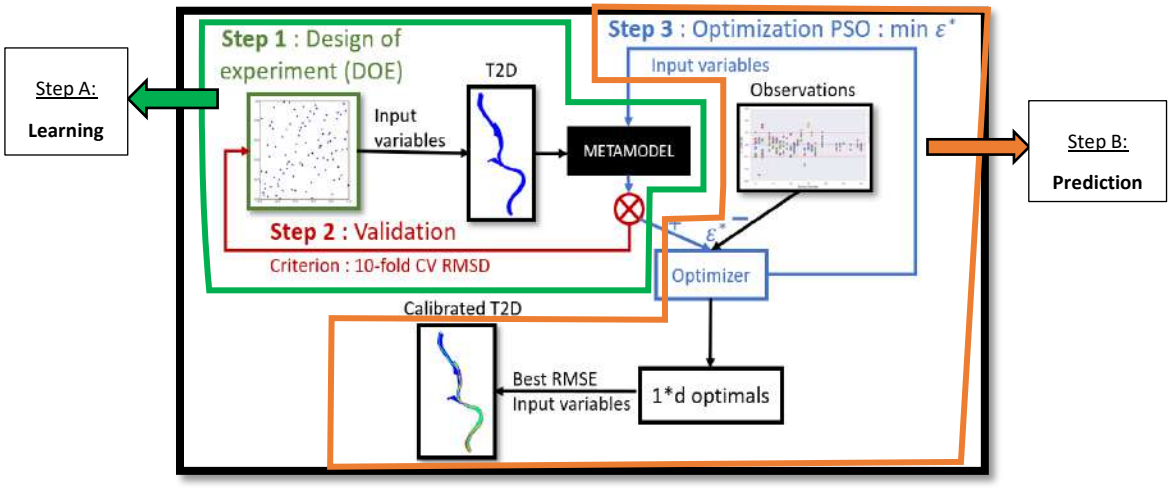


Figure 6: **PYMODAST's** different calibration steps

As in every machine learning algorithm, PYMODAST contains two main stages: The learning step where the problem dataset (x, y) is generated or introduced, then the Prediction step where the best-trained model with the optimized-on-data learning function parameters, is called for extrapolation after being validated.

Each step is leveled by multiples algorithms preceding the next one. Each sub-step will be described and explained in the next parts of this chapter.

4.4.1 Design of experiments (DOE)

To fit the learning functions or the meta-model, inputs or conventionally called "training dataset" is the first step for building the model. The Strickler coefficients sampling strategy is defined by a Design of Experiment (DOE). This sampling procedure ensures the generation of homogeneous and well-distributed data that will be then associated with water levels after the simulation code call to form the training dataset.

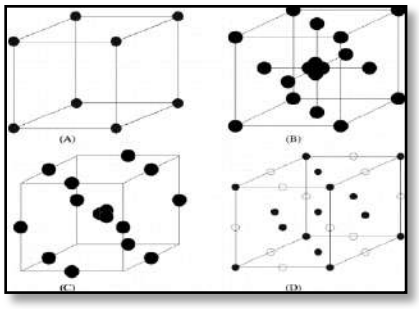


Figure 7: The schematic representation of the design of experiment [7]

For a sake of more vulgarization about DOE, **Figure 5** represents different ways of point arrangement in 3D space. The arrangement patterns are different in their ability to fill this space in which they are, which can be interpreted in a way that there is some points repartition that doesn't provide enough information (if we consider the 3D space as an information container) about the 3D domain space in which they are. Thus, because of this aggregation of some points, some space corners are not covered at all, thus, some information will be missed.

4.4.1.1 Sampling methods

There are multiple methods used to build a DOE, but three common aspects are respected:

1. Factors,
2. Levels,
3. Response.

For our target study cases, classical DOE should not be used to build the input space, otherwise, the DOE would be too large which leads to time costly calculations.

Recent research was conducted by a Korean academic team - a collaboration between the Center for the built environment and School of Civil, Architectural Engineering, and Landscape Architecture at Sungkynkwan University, Suwon – revealed [8] that Latin hypercube sampling method (LHS) is more accurate and efficient for the cases wherein the design parameters changed nonlinearly. On the other hand, the classical fractional factorial design method (FFD) can be a suitable choice when it is certain that there is no nonlinearity effect.

For our problem, classical DOE is not suitable as their size would be too large given the number of variables. For instance, a factorial design would lead to a DOE of size 2^d for d parameters.

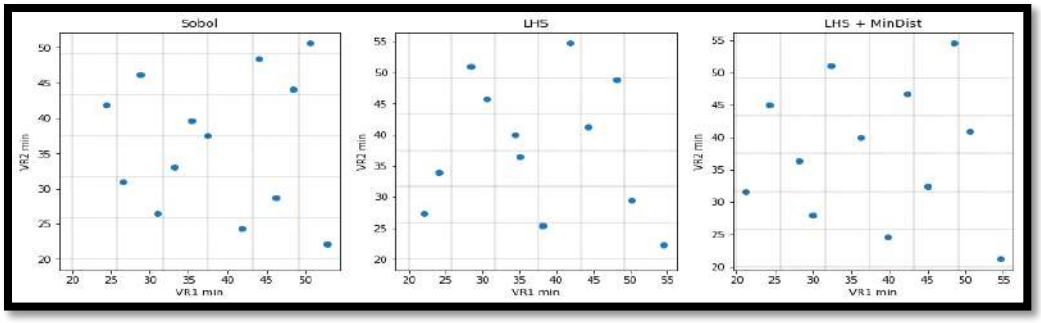


Figure 8: Comparison of DOE sampling methods with 12 points

According to this result, we have used the **LHS method** optimized by the **Mindist**[9] criteria to fill in the most proper way input space and get the maximum amount of information from it.

Once the sampling method of the DOE is selected, the sample size needs to be defined. A value of $\mathbf{10}\boldsymbol{d}$ (with d the number of input variables) is suggested.

NB: In this study, the OpenTurns python library will be used to generate the DOE.

Python code of the sampling method implemented in our code:

```
import openturns as ot

dist = ot.ComposedDistribution([ot.Uniform(0, 1)] * input_length) #input length: number of problem parame

lhs = ot.LHSExperiment(dist, size) #Size : The total generated data number

lhs.setAlwaysShuffle(True)

spacefilling = ot.SpaceFillingMinDist()

optimalLHSAlgorithm = ot.MonteCarloLHS(lhs, n, spacefilling)

samples = optimalLHSAlgorithm.generate()
```

After the DOE creation step, all inputs parameters will be injected into the simulation code (Crue10, TELEMAC-2D …), which will provide the observation (measurements) data. Each data result will be associated with its origin input parameters set generator[10]. Therefore, the dataset (See Appendix 10) is a couple of input parameters combinations with their corresponding results issued from the code simulation.

4.4.2 Meta-model (Surrogate model[11])

It is a learning algorithm (f_q) that constructs a mathematical function based on a dataset, to accurately approximate numerical costly code simulation outputs. The constructed function is trained, then the whole model can be deployed to replace the original simulation code in many applications as we will see for sensitivity analysis, optimization…

[9] Type information optimizer method that, geometrically minimizes the average distance between each two points in the space. Practical example in [18].
[10] Typical simple case is: Multiples Strickler coefficients in different river zones are inputs for the simulation code to generate water level in different river sections. It means with different Strickler combination, different water level values will be observed at the same river sections.
[11] Surrogate == Alternative model that replaces the simulation code.

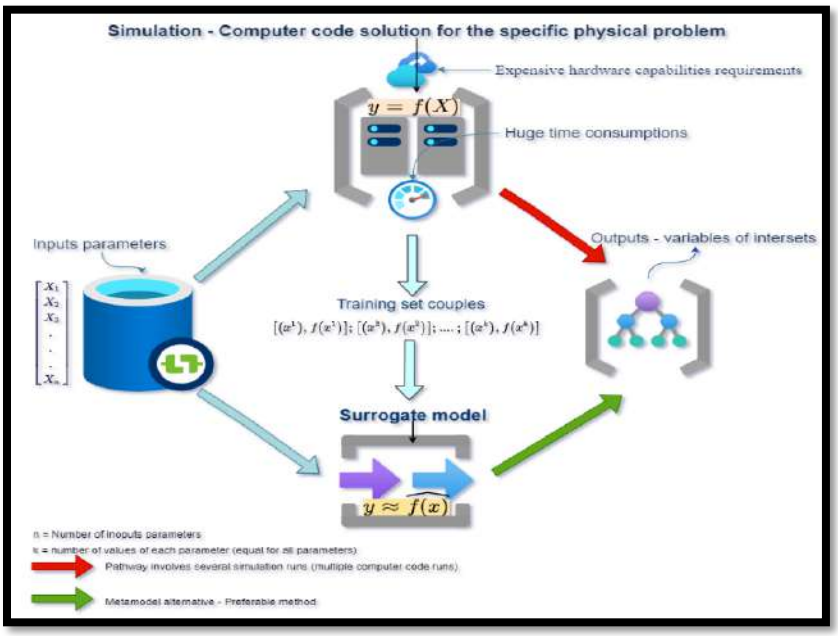


Figure 9: Surrogate model general concept

Machine learning models [12]can be divided into two types:

1. Unsupervised learning
2. Supervised learning

In our case, we are facing a **regression problem** with a supervised learning process**.**

There are multiple supervised learning algorithms. The following were chosen to be implemented in our code:

4.4.2.1 Linear algorithms (hereinafter reglin)

Linear regression is one of the simplest most efficient statistical models. It is a "parametric" model since the estimated function is assumed to be linear[13]. Using a linear model is appropriate in our case. Indeed, deriving Manning's equation (with the assumption of a wide channel) leads to:

$$Z \propto K^{-3/5}$$

Therefore, for example, the linear relationship between friction coefficients and water levels at station *i* can be written as:

$\forall s \in \{1, \ldots, N_s\}, \forall i \in \{1, \ldots, N_c\}$

$$z_i^s = \beta_1^{s,i} {k_1}^{-3/5} + \cdots + \beta_d^{s,i} {k_d}^{-3/5} + \varepsilon_{s,i}$$

$\varepsilon_{s,i}$: Gaussian error term.

[12] Machine learning models is the learning function, surrogate model, or meta-model.

[13] Multiple input exponent values will be used for this study to linearize the input-output relationship for the learning step (in the example equation is $\frac{-3}{5}$. For the coded function see Appendix 11.

4.4.2.2 Regulized regression: Ridge Regression (hereinafter rdg)

It is a normalized linear regression that attributes a shrinking (reducer) coefficient to those less-effect-on-prediction-results inputs. In its conception, it adds a penalization term to the loss function while the model is being trained. One of them is the Ridge regression, and those types of regression were developed for better use of linear regression that can easily get trapped in the overfitting problem[14] without expertise meticulous assistance.

4.4.2.3 Gaussian Process Regressor (hereinafter GPR)

It is a stochastic process that can be used as a machine learning algorithm. It is considered as a Kernel method; therefore, it can predict highly calibrated probabilities through a non-parametric approach by finding a distribution, over the possible functions that are consistent with the observed data. As GPR is a Bayesian method, it starts with a prior distribution and updates it as data points are observed which leads to producing the posterior distribution over functions.

A Gaussian process regression approximates f, a gaussian process is defined as:

$$f(x) \sim \mathcal{GP}\big(m(x), k(x,x')\big)$$

$m(x)$ the mean function of the gaussian process: $m(x) = \mathbb{E}[f(x)]$, $k(x,x')$ the covariance function between *x* and *x'*: $k(x,x') = \mathbb{E}\big[\big(f(x) - m(x)\big)\big(f(x') - m(x')\big)\big]$

The approximation $\hat{f}(x)$ given by a gaussian process regression can be written as:

$$\hat{f}(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x)$$

Where $\alpha_i = k(x_i, x_i) f(x_i)$

$k(x,x')$: Covariance function, called kernel as well.

Two different python libraries were used to perform GPR: Scikit-learn and OpenTurns. Even, the mathematical conception of GPR issued from those machine learning libraries is unique, they differ in the way how the GPR is coded and how the default hyper-parameters values were fixed. For the code see Appendix 12 and Appendix 13.

4.4.2.4 XGBoost

XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate way. The same code runs on a major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples [9].

Discussion:

Those three learning functions were implemented in PYMODAST to be compared according to their prediction's accuracy and execution time. Nevertheless, each model will pass through the validation step to reach out its maximal performance on a specified dataset. Therefore, a validation

[14] Overfitting problem is a usual machine learning trap consists in that the trained model fit the observations as perfectly as possible to minimize the error. However, the model will not perform as well with new data because it is built exclusively for the given data.

algorithm must be developed to conduct this extra training step before going to the optimization step where only one metamodel was retained[15].

It is important to understand that a model behaves differently based on datasets quality, quantity, and type, thus, a certain model can provide accurate results when trained on dataset-A but totally large errors if they are trained on dataset-B.

NB: Above, dataset-A and dataset-B do not refer to any specific data, they are just given for the sake of example.

4.4.3 Validation step

The validation method is the algorithm used to find the most accurate parameters set that conducts the best predictive function. It aims to test the model's ability to extrapolate, avoid problems like overfitting or bias, and give an idea of how the model will generalize to an unknown dataset. Through this, we will be able to estimate how accurately our predictive model will perform in practice. **The K-fold cross-validation[16] (See** Appendix 14**)** has been extensively used by the academic community regarding its accuracy and easiness as it is implemented in almost all machine learning libraries.

When the three models were trained on the BV2016 hydraulic model, the results were as following:

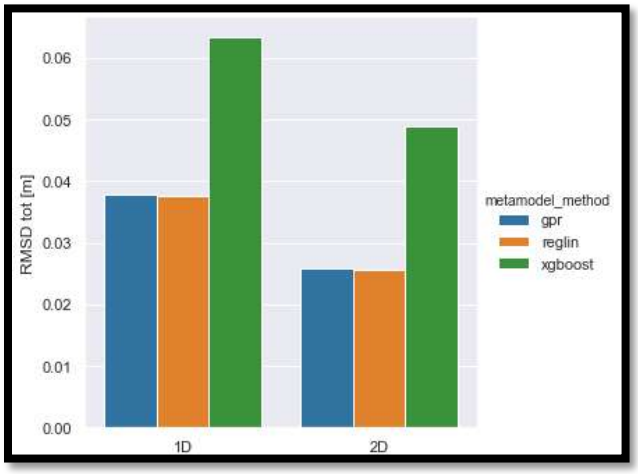


Figure 10: Validation criteria for different metamodels

Meta-models were introduced to simulate 1D and 2D results, and the results, show that they differ in RMSD, which is calculated in the validations step and provide insights about the learning algorithm capabilities to extrapolate out of the given dataset.

Let $\delta_{i,n}^{s} = f_{q_i}^{s}(k_n) - \hat{f}_{q_i}^{s}(k_n)$ be the difference between the metamodel prediction and the hydraulic model prediction for water level at station s, boundary condition q_i, and input Strickler coefficients vector k_n. To measure prediction errors between the metamodel and the hydraulic model, the following scores are studied:

[15] Validation algorithm allows the model to fix its own parameters combination depending to the dataset on which it is been trained. Therefore, the output of this step is a unique function which all parameters have been optimized. This predictive function is defined as the best form of the trained model that it could be provided on that dataset.

[16] K-fold is the number of sets or folder of sub-dataset from the full original (usually K=10)

$$RMSD_{tot} = \sqrt{\frac{1}{NN_cN_s}\sum_{n=1}^{N}\sum_{s=1}^{N_s}\sum_{i=1}^{N_c}\delta_{i,n}^{s}}$$

So, to validate a model, RMSD is chosen as validation criteria. The smaller the RMSD the better the meta-model. Therefore, for example, in Figure 8, linear regression is the best meta-model for our dataset of the BV2016 hydraulic model.

Finally, after that our learning algorithm has found its best form, one step left to predict. It consists of varying, several time[17], calibration parameters in their predefined ranges to find the optimal set that leads to the most accurate prediction. This is the final step, called optimization.

4.4.4 Optimization step

By quantifying the difference between the metamodel algorithm prediction and the real observed values, we will get a distance-like value. Thus, minimizing this distance is the main aim of the optimization step which involves a metaheuristic algorithm to search for global minimum, and among many metaheuristics, the Particle Swarms Optimization algorithm (PSO, see **Appendix 15)** was adopted in PYMODAST. This algorithm has been applied to a bunch of high-dimensional real-word searching problems, it is a population nature-based meta-heuristic optimization technique.

The following parameters of the PSO algorithm are chosen:

- Inertia parameter: $\omega = 0.7$
- 60 particles
- 500 iterations
- $\varphi_1 = 1.2$
- $\varphi_2 = 1.2$

Several repetitions are needed because of the randomness characteristic of the algorithm to obtain a reduced RMSE score or $\hat{f}(k)$ as previously mentioned.

Discission and conclusion:

So far, AOC and PYMODAST were presented from a conception and theoretical point of view. The next section will reveal practical conclusions on those modules, therefore, their quality and performance in front of real data can be inferred and this is exactly what the next sections are for.

4.5 Applications and results

4.5.1 Benchmark AOC vs PYMODAST (Strickler coefficients only)

A benchmark between AOC and PYMODAST was conducted on the Caderousse (CA2020) hydraulic model to compare the efficiency of the previously presented calibration tools.

The optimization results (Strickler coefficients) of AOC are presented in the next figure.

[17] By default, optimization is a repetitive process by its own. In this report, optimization algorithm is repeated several times before choosing the optimal set (Generally: Optimization repetition times = number of calibration parameters).

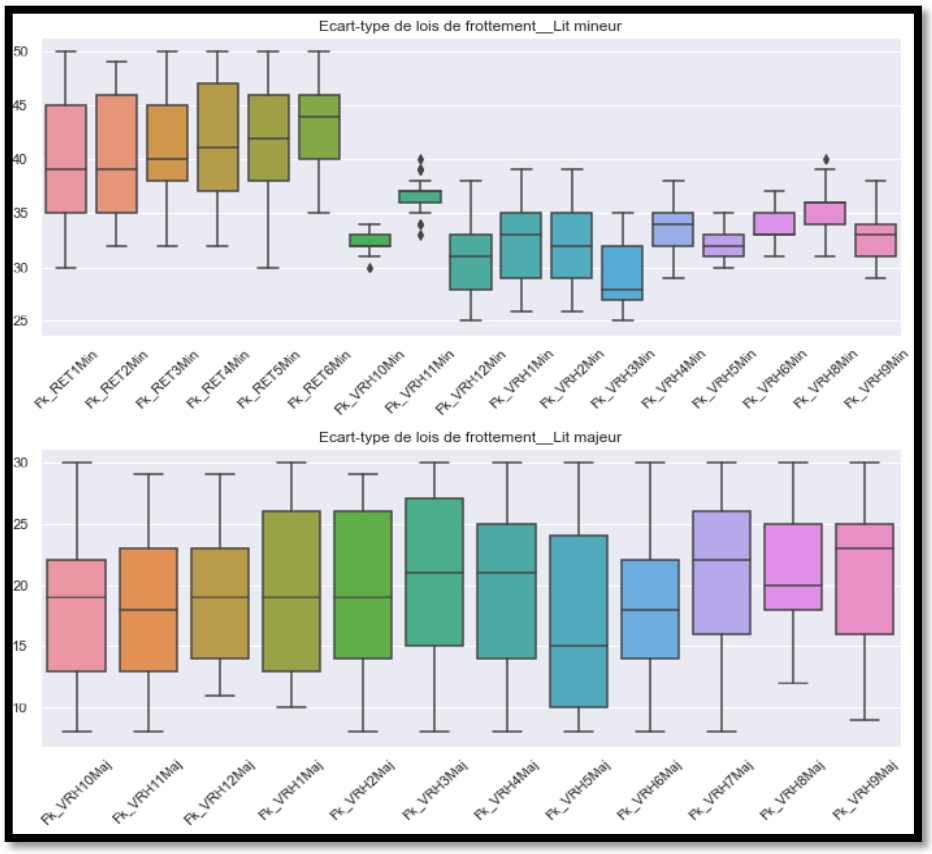


Figure 11: Box plot of the different Caderousse model (CA2020) Strickler zones from **AOC calibration results**

Boxplots are a summarizing type of graphs that give you an idea of how the values in the data are spread out and allow you to display the quartiles (25%, 50%, and 75%) and the outliers.

The variability of the data and the observed dispersion come from the fact that AOC (same observation on PYMODAST) implements a metaheuristic algorithm (Simulated annealing method in the case of AOC), which gives some randomness in the results. Those boxplots show that we get a different optimized set of values of Strickler zones (29 in this model) after each iteration, especially for the floodplain.

We can observe a contrast between friction zones of the channel bed of the Vieux-Rhône[18] and the upstream part. See Appendix 9α.

- From **Fk_RET1Min** to **Fk_RET6Min**: channel bed from section 200.7 to section 212
- From **Fk_VRH10Min** to **FK_VRH9Min**: channel bed from section 189.4 to section 200.5

We can interpret the floodplain friction zones values dispersion zones by:

1. Some floodplains frictions zones are not critical for this model calibration process. As those values are dispersed and the median values (the horizontal line of the box) do not have a specific pattern to describe skewness.
2. This dispersion means that it will be better to fix those values with engineering expertise to gain more time and accuracy for further calibration of this model.

[18] Vieux-Rhône ➔ VRH

3. It can be interpreted as the friction configuration in the floodplains cannot be reduced to a simple coefficient value and this generates a large dispersion because of the multiplicity of values that those coefficients can take.

Large friction zones data dispersion in the floodplains part because we have injected those values even when we are not pretty sure that they will affect the water level profile and help the calibration process. It means, that the friction zones profiles that we have chosen will be less dispersed provided that the water level reaches the floodplains.

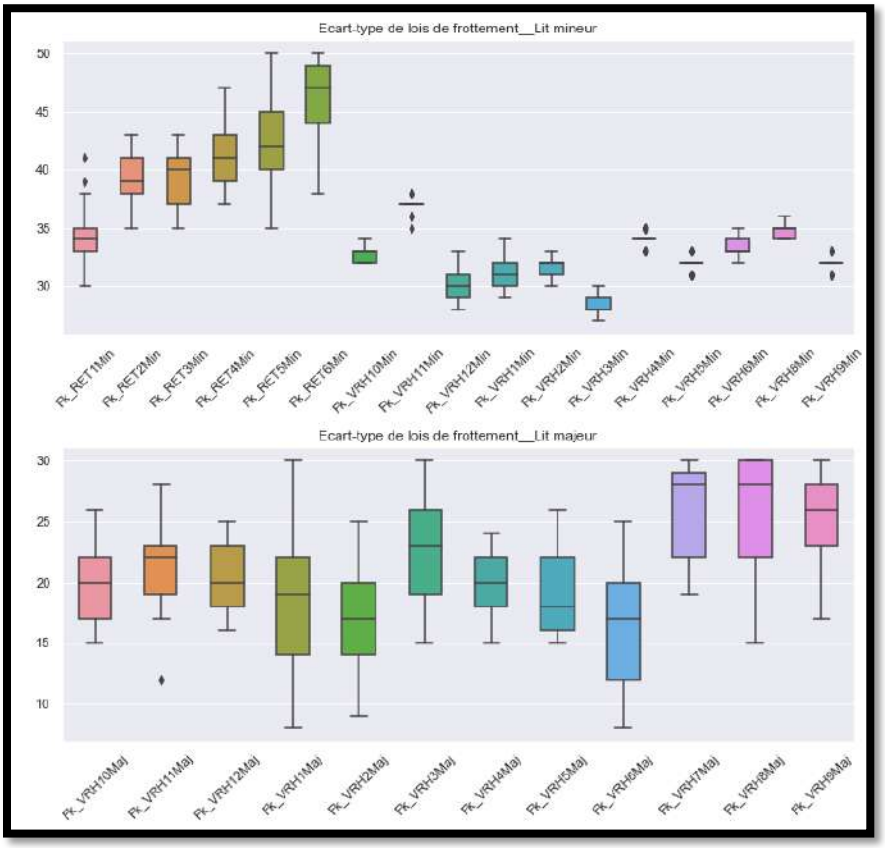


Figure 12: Box plot of the different Caderousse model (CA2020) Strickler zones from **PYMODAST calibration results**

Globally, less dispersion can be observed for PYMODAST repeated optimizations. Some friction zones of the Vieux-Rhône were found to have the same coefficient after every single optimization iteration. Even with a large friction interval, PYMODAST could fix several values of those Strickler coefficients. Thus, those frictions values will be fixed for further studies of this model. PYMODAST could provide more accurate channel friction values.

Less dispersion is also observed in the floodplains frictions zones.

General conclusion:

Observation and interpretation that we could infer from the AOC and PYMODAST calibration studies are that they provide the same dispersion pattern. This is because they are based on the same physical relationships and natural conditions configurations. However, PYMODAST excels

AOC in the calibration process and it can provide more accurate calibration combinations[19] for further studies.

Discussion:

Previous observations and built-on conclusions around AOC and PYMODAST are not scientifically established, which means that the interpretations may vary because there is no normalized framework to unify inferences about their patterns and results dispersion. This is was the root of critical thinking which turns out to a parametrization proposition of the problem. It will allow the quantification of this observed dispersion and provide insights for further studies as it allows to handle more details about the available data.

The next sections are dedicated to calibration tools parametrization and how they could be beneficial in quantifying calibration tool's efficiency and accuracy.

4.5.2 Introduction of two new metrics to compare AOC vs PYMODAST

To assess the code performance thoroughly and quantify its stability, we have introduced two metrics that quantify the stability and accuracy of AOC and PYMODAST calibration tools[20].

In a single calibration, PYMODAST/AOC provides RMSE value for every single flow configuration then, we have developed a code to compute new metrics based on RMSE values.

NB: *Those two metrics do not reveal the learning model capabilities, in contrast, they precise the efficiency of the calibration procedure.*

See Appendix 16.

4.5.2.1 The robustness

Robustness is the ability of the model to consistently provide accurate outputs across variable definitions and functional forms. Therefore, a model is robust if it sustains its performance while the variables or parameters assumptions are modified.

In our case, PYMODAST / AOC robustness has been quantified by optimizing on multiples water surfaces profiles of a hydraulic model.

The robustness formula (for a given iteration i) is:

$$\text{Robustness}_{(i)} = \frac{\sqrt{\frac{\sum_j^N \left(RMSE_{(i,j)} - \overline{RMSE_i}\right)^2 \cdot \alpha_j}{\left(K_{ponderations} - 1\right) \cdot \sum_j^N \alpha_j}{K_{ponderations}}}}{\sigma_{Z\_measurements}}$$

With:

- $\boldsymbol{RMSE_{(i;j)}}$ **:** Root mean square value for the water surface profile number j for iteration i
- $\boldsymbol{\overline{RMSE_i}}$ **:** mean of individual $\boldsymbol{RMSE_{(i,j)}}$ over all water surface profiles (for iteration i)
- $\boldsymbol{K_{ponderations}}$ **:** Number of non-null ponderations values
- $\boldsymbol{\alpha_j}$ **:** j[th] water surface profiles ponderation values

[19] In the Caderousse model, we get 29 friction zones values after each optimization process iteration

[20] Each tool was tested according to its previously introduced RMSE formula, i.e. AOC objective function is without ponderation coefficient.

- $\sigma_{Z\_measurements}$ **:** Standard deviation of the water level measurements (Usually fixed to 10cm)

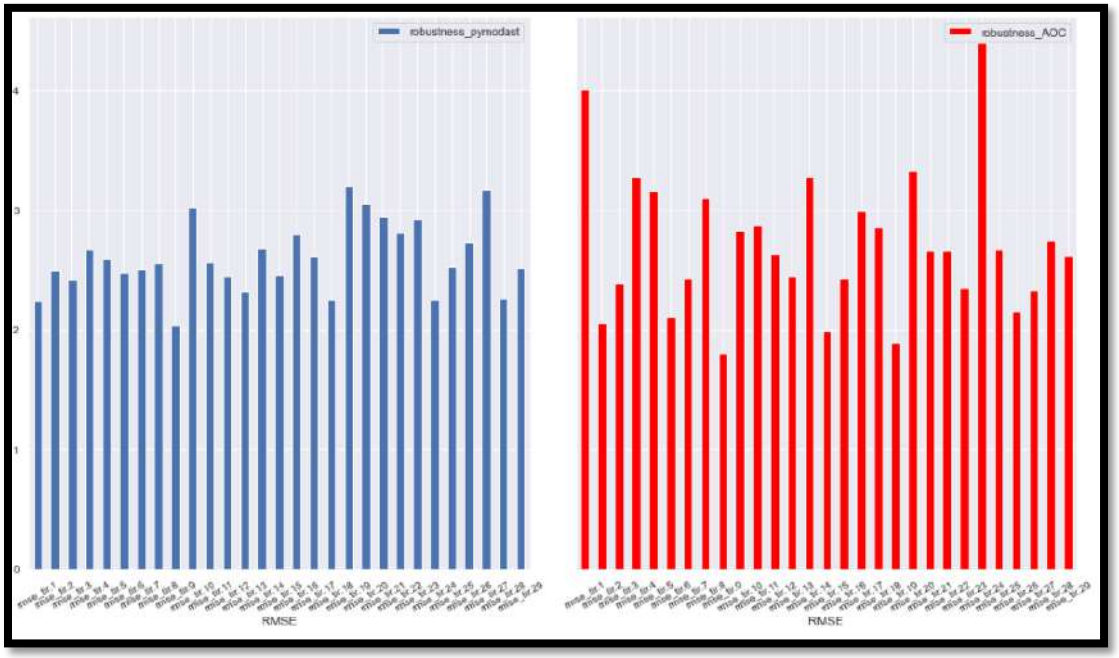


Figure 13: Bar plots of the robustness metric in Caderousse 2020 model (CA2020)

Interpretations:

PYMODAST can cope with changes (*different water surfaces profiles* ➔ *Different flow conditions*) which means that PYMODAST is more stable than AOC. PYMODAST contains fewer extreme values comparing to AOC, which makes it more robust through repetitions.

4.5.2.2 The repeatability

In environmental science, the most interesting question to which the researcher asks is: what is the possibility to repeat the same experience[21] and get the same results? To find the answer, the researcher conducts multiple times the same experiment maintaining the same condition and perform statistical analysis on the results.

Repeatability formula (for a given water surface profile j):

$$\text{Repeatability } (j) = \frac{\sqrt{\frac{\sum_i^N \left(\boldsymbol{RMSE}_{(i,j)} - \overline{\boldsymbol{RMSE}_j}\right)^2}{N_{RMSE}-1}}}{\sigma_{Z\_measurements}}$$

With:

- $\overline{\boldsymbol{RMSE}_j}$ **:** mean of individual $\boldsymbol{RMSE}_{(i,j)}$ over all iterations (iteration number was fixed to be equal to Strickler values number)
- N_{RMSE} : Number of roots mean square errors

[21] Here, same experience means that the optimization algorithm is repeated several times with the same hyper-parameters initialization and configuration to be able to compare results.

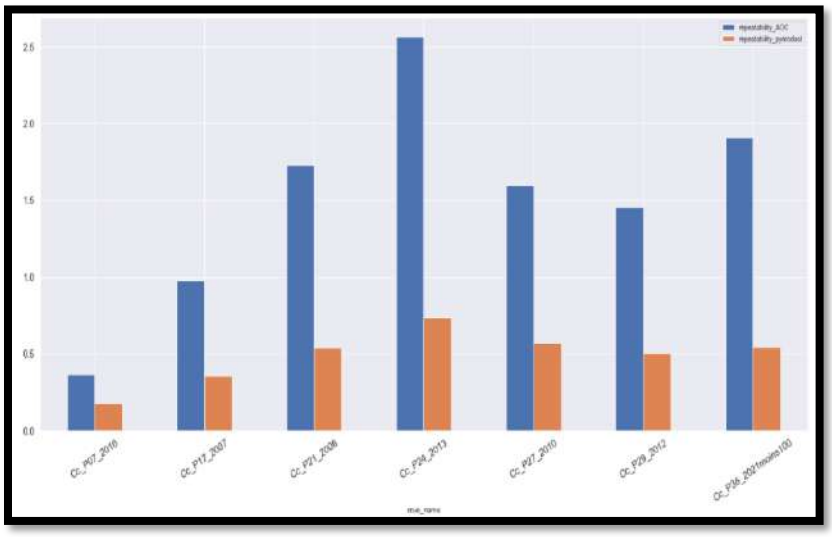


Figure 14: Bar plots of the repeatability metric in Caderousse 2020 model (CA2020)

Interpretations

As repeatability is normalized by the instrumentation error $\sigma_{Z\_measurements}$, therefore, if the value is less than the unit, we can conclude that:

- The equifinality problem will not be considered as multiples resulting optimization parameters set are closer to each other. This is observed just in the case when PYMODAST is used.

The equifinality problem in the AOC optimization algorithm, as its repeatability is **>1.** This means that it provides different optimal sets after each repetition; therefore, many parameters' combinations values can lead to an optimum.

Discussion:

At this level, the previously mentioned calibration tools were implemented with a single calibration variable (Strickler coefficients defined by zones). However, theoretically, our goal is to introduce more physics to this purely mathematic-process toolbox by injecting more features into the problem learning function. On the other hand, the latest research on machine learning has shown that adding more features to the problem is not always beneficial, on the contrary, it can overfit the learner[22]. Adding more features arisen from hydraulic domain knowledge should be engaged with more advanced methodology and scientific arguments based on *Features engineering science*.

In the next section of this part, the discharge coefficient will be added as a problem feature and results will be analyzed thoroughly.

[22] Learning function or the learning algorithm

4.5.3 Strickler and discharges coefficients calibration

4.5.3.1 Issue source and causes - procedure trigger (Calibration challenges over the Vieux-Rhône hydraulic models' part)

The principle aim of the calibration process is to find the frictions coefficients best set that minimize the water levels prediction errors. In the modern optimized process as PYMODAST, the problem is breaking down to a mathematical problem of optimum research.

After many model calibrations projects at CNR, many studies revealed that the predicted water levels values on the **VRH** part exceed the fixed margin of error (usually 10 cm) due to multiples factors (changes in the channel section, changes in the main relation of the rating curve in hydrometric station). Therefore, theoretically, the option of injecting more flexibility into the problem will turn out on a learning process with more explorative capabilities and with enough information about the physical relations behind it.

In the process of calibrating hydraulic models, there are water levels profiles that are difficult to calibrate (large error between measured values and values predicted by the metamodel and the optimization process). This is because the flood discharge set for a water levels profile is not the best choice among the flows that fall within the margin offered by the measuring stations. How to integrate this constraint into problem parameters and optimize it to improve the calibration of this line without causing destabilization of the water levels profiles that are calibrating well?

4.5.3.2 Methodology

We have proposed a method that consists in causing a perturbation on the model, it means increasing or reducing a discharge in one or more of the model's nodes for a specific water level profile that fits well and analyzes the behavior of PYMODAST predicted results. Before that, giving a fixed quantity ΔQ will create a new input feature interval, where the optimal multiplicator coefficient Q_{factor} will be found by the optimization algorithm to correct the previously changed discharge (the caused perturbation). This new input will be considered for the simulation code numerical solution (more inputs for the DOE) preceding the learning step. For details about this methodology, schematic see Appendix 17.

4.5.3.3 Case 1: no perturbation

In this case, we do not change the discharge value in any of the model's nodes, but we introduce flexibility ($\Delta Q \neq 0$) on Q_{factor} value interval to let the optimization algorithm provide a value in [1-ΔQ; 1+ΔQ]. Therefore, the problem will get more inputs as much as we add flexibilities on nodes discharge value. See Appendix 3 and Appendix 5.

CE2016 model has been chosen to assess PYMODAST optimization performance on more features. Here we have adopted three different AOC files, with different flexibilities that we have launched our calculation.

- **File A**: The original file, without any modification on ΔQ
- **File B**: Introduce one modification on ΔQ =10% in **Nd_RET151.520** [23] for **Cc_P009_LE20160426** water surface [24]
- **File C**: Introduce modifications on both model's surface profiles node configuration (ΔQ =10% on the corresponding node)

[23] The very upstream CE2016 hydraulic model's node. See Appendix 6
[24] CE2016 contains two water level profiles : Cc_P009_LE20160426 and Cc_P012_LE20160202

To quantify successive launch of PYMODAST results in dispersion on the same starting configuration and quickly generate it and compare it with other data, the calculations were repeated 5 times for each case (files), and box plots were plotted to visualize the dispersion.

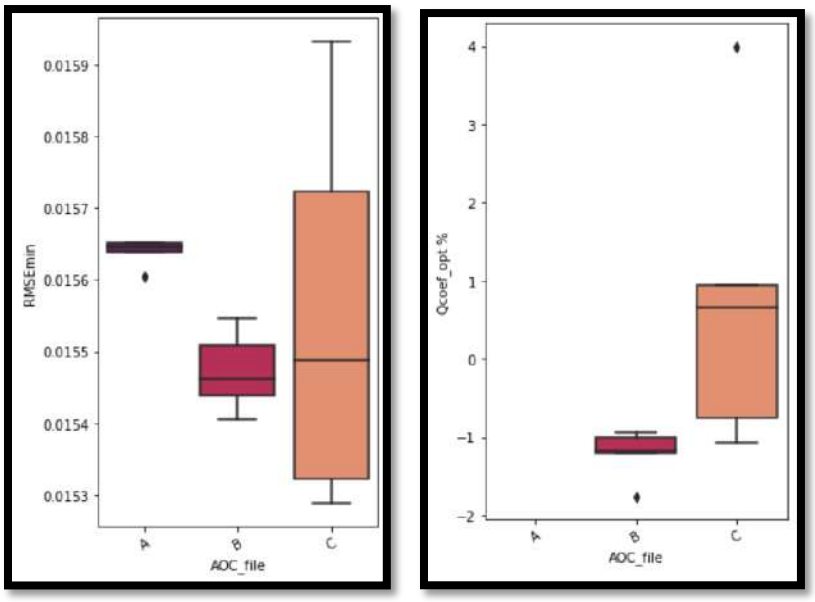


Figure 15: **a)** $RMSE_{min}$ dispersion results **b)** Optimal Q_{factor} provided by the optimization step

In Figure 15.a, even with a difference of 10^{-4} meters order of magnitude between $RMSE_{min}$ we observe that we enhance the $RMSE_{min}$ by adding more features to our learning algorithm, which means that the more information we provide about the physic ruling behind the problem, the learning algorithm can globally extrapolate better.

Figure 15.b shows that when we increase flexibility on discharge prediction, PYMODAST provides - after each run -the optimal value Q_{factor} (between -1 and 1). Thus, according to PYMODAST predictions, increasing the discharges in the corresponding node **Nd_RET151.520** for both water surface levels will reduce the $RMSE_{min}$, therefore, enhance the prediction process.

4.5.3.4 Case 2: two perturbations

Here, in the same model, we have conducted another experiment. Adding two flexibilities, we have modified the discharge with different ratios in the corresponding node (in DCLM files) and train three different models five times to assess the dispersion of the results.

Three different configurations were set up as:

- Study *CE2016* **<u>where</u>** *CE2016_orig_* $Q \cdot (\pm \boldsymbol{Q_{attributed\_factor}})$: As CE2016 is a relatively simple model with two water surface levels, discharges in **Nd_RET151.520** were modified for both (WSL) as follows:

  One. First water surface level Cc_P009_LE2016**0426**: the original discharge value in **Nd_RET151.520** was: 432 m^3/s. Engineering expertise estimates that this discharge should be bigger. Thus, **a positive** $\boldsymbol{Q_{attributed\_factor}}$ was attributed to this node which turned into:

$$Q_{\textbf{Nd\_RET151.520 of Cc\_P009\_LE2016}\textbf{0426}} = 432\left(1 + Q_{attributed\_factor}\right)$$

Two. Second water surface level Cc_P012_LE2016**0202**: the original discharge value in **Nd_RET151.520** was 541 m^3/s. Engineering expertise estimates that this discharge should be smaller. Thus, a **negative** $Q_{attributed\_factor}$ was attributed to this node which turned it into:

$$Q_{\textbf{Nd\_RET151.520 of Cc\_P012\_LE2016}\textbf{0202}} = 541\left(1 - Q_{attributed\_factor}\right)$$

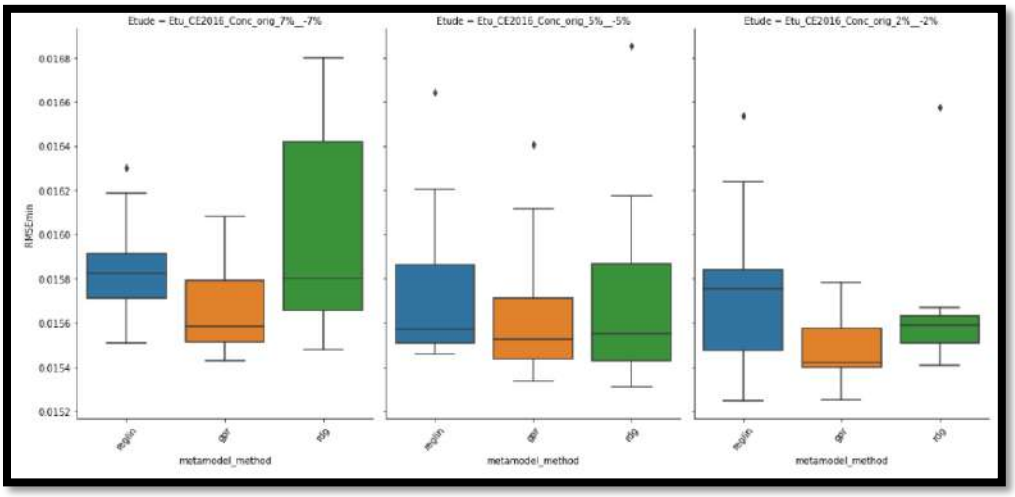


Figure 16: The $RMSE_{min}$ Box plots (for 3 different $Q_{attributed\_factor}$ ±7%, ± 5% and ± 2%)

Observation and conclusion:

- Large dispersion is observed and there is no pattern between different experiences
- The Gaussian process is better in prediction as it provides the smallest median error each time over the three experiences.
- It seems that $RMSE_{min}$ dispersion is in a negative correlation with the discharge ratio.

Even if the $RMSE_{min}$ median value is physically acceptable, however, we cannot conclude anything about the models.

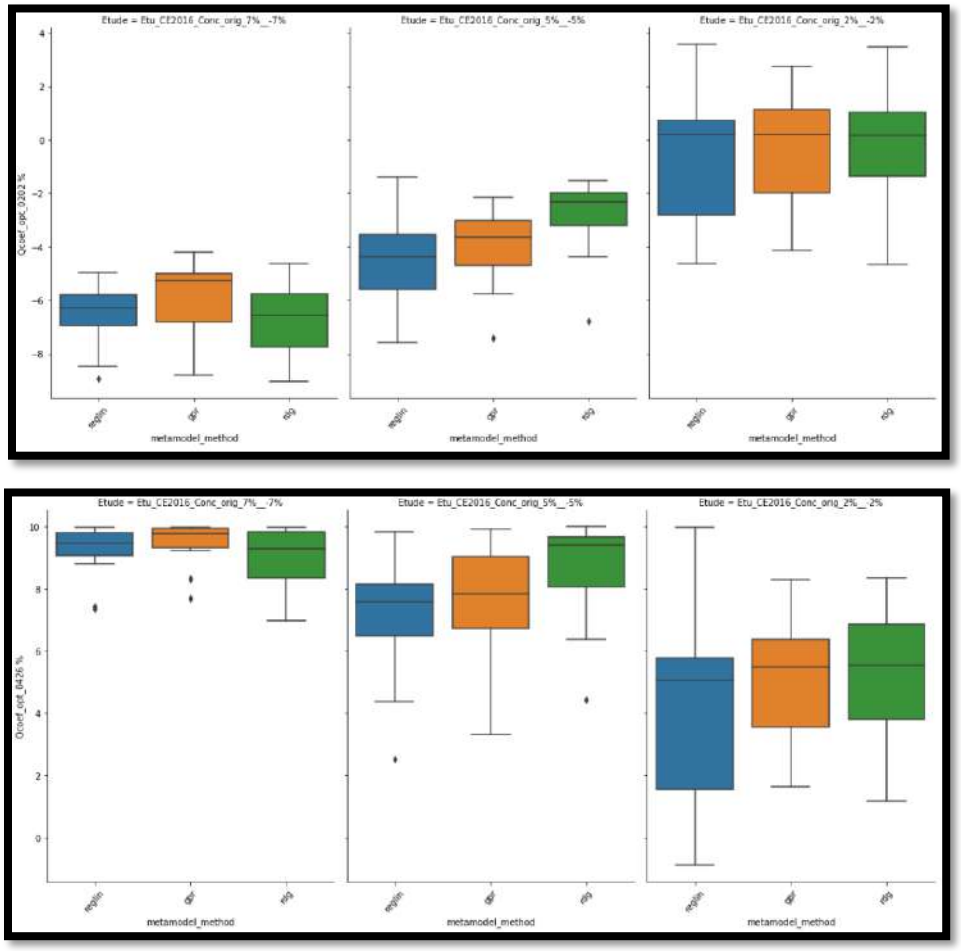


Figure 17: $Q_{predicted\_factor}$ prediction results Box plots

NB: $\boldsymbol{Q_{factor\_predicted} = 1 - Q_{coef\_opt}}$

Appendix 18 presents the data categorization scale into "Dispersion" and "Prediction" characteristics of each configuration and reveals how the data are clustered and provides some pattern insights.

Discussion:

No conclusion can be derived from Appendix 18 about the model performance or predictions improvement besides that the higher the discharge coefficient the better the prediction and the lesser the dispersion of the results.

4.5.4 Conclusion

The calibration process relies on building a metamodel from a Design of the Experiment. The calibration problem is then solved using a particle swarm optimization algorithm. In order to reduce computational costs, the loss function to be minimized is estimated by the metamodel. This procedure is time-saving since metamodels run faster than classical hydrodynamic models. However, it seems hard to derive a conclusion on calibration process performance improvement when we added more features, especially that we have used a simple hydraulic model (CE2016), and the only remarkable result is that when we add the discharge coefficient, we globally get smaller error values but more dispersion over the calculations.

5 Sensitivity analysis

5.1 Introduction

Clear definitions, assumptions, and hypotheses are needed to start any modeling problems, and if this is unclear, then the rest will be unclear, imprecise, ambiguous, and confusing. Those assumptions and simplifications are mainly made on input data, may include uncertainties that have an impact on the model response. Therefore, to establish a scientific-based methodology that can quantify those uncertainties and determine confidence intervals, probabilistic engineers conduct the uncertainty propagation and sensitivity analysis study to quantify the input-output relationship in a probabilistic framework, which will allow them to validate their computer code or provide them arguments on system design safety, and this is the core idea behind any sensitivity analysis study.
Sensitivity analysis can be either local (LSA) or global (GSA). GSA was established to overcome the limitations of local methods (linearity and normality assumptions, local variations…). In contrast to local sensitivity analysis, it is
referred to as "global sensitivity analysis" because it considers the whole variation range of the inputs [10].

All coming definitions and applications were conducted according to GSA methodology which it relies upon:

1. Mathematical definition of the simulation model.
2. Uncertainty sources identification and precision of the variable of interest.
3. Probabilistic description of the uncertainty sources around inputs parameters by attributing a PDF distribution to each input variable (PDF can be inferred from the calibration or engineering expertise). See Appendix 19.
4. The uncertainty propagation through the model using probabilistic sampling methods available on different software (Monte Carlo, Quasi-Monte Carlo…) and multiples model calls.

Then, following previous steps, several possible schemes may be applied based on the targeted probabilistic variable of interest (statistical moment, distribution, failure probability…) and the aims of the study.

In this report, the uncertainty propagation steps, and methodology are presented in Appendix 20. The steps are the following:

- Step A: Consists in the model definition and probabilistic variable(s) of interest. Similarly, it is the same as defining a deterministic model, we define the parameters of the function and the response quantity which will be provided by the function.
- Step B: Precise and quantify uncertainties origins in each input parameter and attribute a PDF to them. The final problem will be multiples sets of inputs (combination of inputs random values) that can be represented by joint probabilities or copula that describe dependence among variables.
- Step C: It is the uncertainty propagation step. In most cases, a sensitivity analysis "Step C' "is required to assess the impact and weight of input parameters onto the randomness of the output.

Finally, we can summarize the SA as follows:

The objective: Evaluate the importance of the input parameters of a model on the output variables of interest.

Why? This allows us to look at the parameters on which to focus to decrease the uncertainty associated with our output variable.

Results: The sensitivity analysis makes it possible to answer questions such as:

1. What are the most impacting input parameters of my model?
2. How much of the variability in my model is explained by the input parameter X?
3. How does the variability of the input data impact the variability of the model's output response?

5.2 Tools

To conduct a GS, the simulation code must be coupled with a sensitive analysis framework. Multiples tools and software are available out there, thus, a compatible choice should be made based on the study interests. In our case, two available options in the box:

5.2.1 Prométhée-Crue

Prométhée is a computing environment developed by IRSN[26]. It makes it possible to carry out many calculations within the framework of the numerical modeling of uncertainties. It implements the statistical methods of the R language. It makes it possible to define the variant parameters and their statistical distribution and to launch a large number of Crue9 simulations. It integrates traceability into the parametric modeling process.

5.2.2 Persalys: A GUI built-on OpenTurns

OpenTURNS (OT) is a Python library built on C++ for advanced Global Uncertainties Processing Methodology developed by the EDF R&D Uncertainties Network. It offers a large set of methods for quantifying, propagating, and prioritizing uncertainties.

- Uncertainty quantification
- Uncertainty propagation

5.2.3 Persalys-Crue10 coupling

I general, the coupling means assembling two or more individual functionality embedded in two different systems. In sensitivity analysis and using tools that we dispose of; a coupling method will be possible following the workflow presented in Appendix 21.

Coupling an external simulation code directly into Persalys is possible if it can communicate with the code via input and output files and that it is executable from a terminal or in batch mode (the case of the Crue10 code). A physical model is specially developed for this purpose.

There are two different ways to couple Persalys with an external code:

- Using the GUI through "**Modèle de couplage** "window. See Appendix 22 A, B, and C. To be able to conduct this coupling methodology, the following steps should be respected:
    1. Creating template input file containing tokens instead of the values to be varied.
    2. Execution of the external code which will take inputs from the template file.

[26] Institut de radioprotection et de sûreté nucléaire.

3. Reading the results in one or more output files written by the external code.

- Developing Python code (Crue10 code or functions) using the Python console through " **Modèle Python** ". It means introducing the code manually inside Persalys Appendix 23.

5.2.4 Persalys VS Prométhée_Crue

Biographical research revealed many similarities between the previously SA software, and Appendix 24 summarizes the embedded methods and algorithms in each one.

NB: Next coupling analysis will be carrying out on Persalys software.

5.3 Sensitivity analysis in 1D models

5.3.1 Case study SB2013

As part of the updating of the *Sault-Brénaz 1D hydraulic model*, calculations of water surface profiles for the design flood must be carried out during the verification of the constraints imposed on the concessionaire via the design brief. The computation conditions of the *constraint and consideration 2.1*[27] (respect of a 50 cm freeboard on the unsinkable embankment during the design flood for a nominal operation of the development) were retained and two distinct scenarios were studied, corresponding to the two possible rating curves at the downstream dam. See Appendix 25

The main source of uncertainty: The simulation results obtained are therefore based on the strong assumption that the hydraulic setting parameters (Manning -Strickler coefficients in particular), calibrated for the observed floods, remain the same for higher discharges (floods), and up to the project designing discharge. Thus, questions intuitively raised on extrapolation possibilities of calibration parameters, therefore it became necessary to determine the level of confidence in these results nominal calculations vis-à-vis the inherent uncertainties in modeling (calibration).

Study objectives: The primary objective of this study is to determine the degree of confidence in the results of the nominal calculations for the *constraint and consideration 2.1*.[28]

Tow possible scenarios are distinguished according to the rating curve used to determine the downstream dam level in the calibration process:

1. **Cusset's EGR** mathematical model (CU)
2. Curve extracted from the volume of the DE presenting the results of the **physical model of Gerland** (MP)

The nominal value of the dam downstream level can take 2 values, depending on the scenario studied for a plant flow rate of 580 m^3/ s:

1. **201.41 m NGFO** for **CU**
2. **202.53 m NGFO** for **MP**

NB: All next steps will be conducted on the **MP model**.

[28] **Constraint 2.1**: the embankments that are unsinkable at the project designing discharge of the reservoir must have a clearance of at least **0.50 m** in relation to the water surface profile corresponding to **3 150 m^3/s which is the project designing discharge**.

5.3.1.1 Uncertain parameters

Uncertainties can be differentiated according to their nature. Epistemic uncertainties are linked to the imprecision of the information, due to a lack of knowledge resulting from errors measurement or relating to a non-justified expert's opinion. In contrast, random uncertainties are related to the natural variability of the parameter.

The first step to conduct a sensitivity analysis on the SB model is determining the parameters that are the source of uncertainty in the model on a set of parameters chosen from among the most influential then quantifying the ranges of variation and choosing an appropriate distribution law to model the best the variability of each of these variant parameters. Secondly, the uncertainties will be propagated by a sampling method, to have results.

Table 3: Uncertainty parameters PDFs

Parameter	PDF – Modelling the parameter variability
$\mathbf{Q_{factor}}$ (for hydroelectricity production groups. See Appendix 27)	Triangular distribution (0.85,1,1.15) ➔ derived from the triangular distrib. Of (- 493 m^3, -580,667 m^3)[29]
C : Dam's weirs coefficient (Coefficient De Marchi)	Normal distribution ➔ N (0.84, 0.05)
$\mathbf{Z_{aval}}$: Downstream's dam water level	Normal distribution ➔ N (201.41, 0.125)
K1 [30] **:** Modulation coefficient of the Strickler coefficient tq: $K_{total} = K_{nominal} + K1$	Normal distribution ➔ Normal N (0, 2) + **truncation**[31] (Appendix 26)

5.3.1.2 Variables of interest

In sensitivity analysis, we can choose many types of the variable of interest that are supposed to provide insights about the model future potential enhancement or contribute to input-output relation understanding. In our case, as we are dealing with a validation problem (validation of constraint 2.1 fulfillment), our variable is a response variable that describes a physical phenomenon (water level, flood discharge…) under a set of combined parameters conditions.

Whatever the analysis conditions in both dam downstream level cases, the point where the freeboard remains the lowest is at the **P67.200 cross-section profile**, therefore water level variability in this point will be analyzed as the **P67.200** profile was selected as the most representative point of the problem. See Appendix 27.

5.3.2 Results

5.3.2.1 Computation time

When we deal with a large amount of data, the main priority is to reduce calculation time to perform as much as possible of different studies that will provide insights and a clear conclusion.

In our case, we compared the execution time of the code in two different ways and it turns out that "**Modèle Python**" is 2 times faster than the GUI coupling "**Modèle de couplage**".

[29] Discharge is negative because it is a substruction to the total discharge streaming in the river.

[30] The uncertainty range of the Strickler coefficient depends on the flow, and the calibration data from common operating flow rates.

[31] A probability distribution for a random variable *X* is said to be truncated when some set of values in the range of *X* is excluded. This was introduced as many runs resulted as errors.

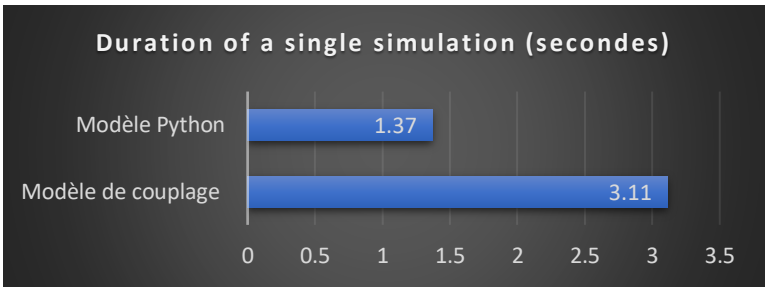


Figure 18: Calculation time duration comparison of a single Crue10 simulation using the Persalys GUI and Persalys python console

Therefore, all other steps will be built on the algorithm developed in Persalys python console with input parameters configuration and probabilistic characteristics of inputs detailed in Appendix 28.

5.3.2.2 Sensitivity to the design of experiments

To quantify the dispersion of the results, we have performed a simulation for multiples DOE sizes. Results are shown in Appendix 29. While Appendix 30 shows the probabilistic distribution of the variable of interest resulting from the DOEs. PDFs show that the study case won't be sensitive to DOE size as the PDFs have the same shape and the same probabilistic characteristics.

5.3.2.3 Design of Experiment:

In the conclusion of the [11] scientific article, the *Quasi-Monte Carlo sampling method* was recommended to lower the computation time. Based on that, an uncertainty propagation study following the Quasi-Monte-Carlo approach with *2000 samples* was conducted. It consists in carrying out a random selection for each of the input variables identified as a source of uncertainties, then executing the hydraulic simulation code associated with each of these 2000 sets of parameters. We can then statistically analyze the results of the 2000 calculations obtained and know the amplitude of variation of the results.

5.3.2.4 Impact of each uncertain parameter on the response variable -- Variance based sensitivity analysis:

The objective of the GSA presented in this chapter is to quantify the impact of each factor on the variability of the results extracted from the P67.200 profile for which the freeboard is minimal (where the constraint should be verified).

The interest here is to classify the factors to see which one has the most influence on the output, thus determine the one that will require the most precision and then reduce the variability of the output.

Several methods exist to perform this analysis, **Sobol indices** by Saltelli [12](Appendix 31) method which is embedded in Persalys as the main method for the global sensitivity analysis. Individual and total Sobol indices were calculated for this study. The Persalys's Sobol computation configuration is in Appendix 32, and the results are presented in Appendix 33 which shows that the water level at P67.200 is mainly impacted by the Manning coefficient modulation.

Discussion

After performing a coupling model, Appendix 36 contains all the resulting probabilistic characteristics of the variable of interest. Following interpretation and conclusion can be derived:

The mean value is so close to the nominal value which means that the model behavior is linear in this scenario (MP model conditions). The total extent of the calculation results at P67.200 is 36cm. The maximum value reached, Z (P67.200) = 204.036 mNGFO, is less than the dimension of the top of the jumper $Z_{freeboard}$ = 204.37m NGFO. This is also the case for the other points of the section for which the results have been extracted. Moreover, in 90% of the cases the water level at P67.200 is inside [$\mathbf{Q_{5\%}}$ = 203.833 mNGFO, $\mathbf{Q_{95\%}}$ = 203.949 mNGFO], thus after those uncertainty propagation results, water level at P67.200 for this scenario is between ($Z_{freeboard}$ - $Q_{5\%}$ = +53.7 cm ; $Q_{95\%}$ - $Z_{freeboard}$ = - 0.421 mNGFO).
For the nominal case, the water level below the dyke crest (freeboard) is +53cm. Therefore, freeboard largely encompasses the upper margin of error on the nominal case. It means that in this scenario the constraint 2.1 is respected.

5.3.2.5 Metamodel-based global sensitivity analysis: The Gaussian process

After proving that GSA is a powerful approach to determine the key uncertain input parameter that impacts the uncertainty of model output predictions. So far, Sobol indices are based on Quasi-Monte Carlo simulation, which is time-consuming in the case of computationally expensive models (Crue10). In this respect, a surrogate model or the so-called metamodel such as the Gaussian process (See Appendix 38) has gained attention in the last decades, as they provide results in negligible time by replacing the original simulation code. Then, this metamodel will be the main basis for Sobol indices determination.

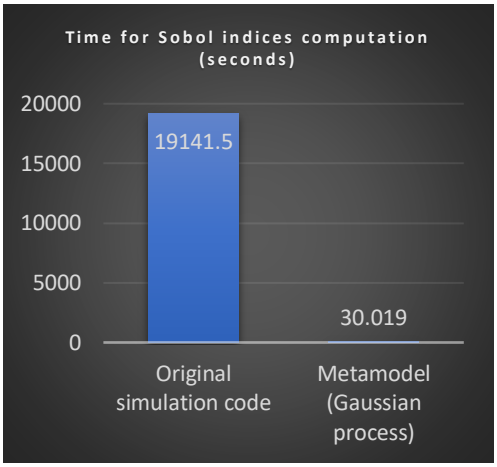


Figure 19: Sobol indices' time computation comparison

Moreover, in the Gaussian process, the same procedure was repeated to propagate uncertainty through the model and shows PDFs issued from metamodel computations. Appendix 37, shows the same table of probabilistic characteristics of variable Y3, which is logic as the metamodel was built on the simulation (Quasi-Monte Carlo) code results.

6 Conclusion

In the considered scenario (MP), the water level at the dike crest (204.37mNGFO at P67.200) is not exceeded in the 90% confidence interval. 100% of the results of the MP scenario are even under the $Z_{freeboard}$-10cm dimension. Therefore, the freeboard of the water line on the embankment seems sufficient to absorb these uncertainties. In this scenario the Villebois's dam is operable[32] over the entire panel of calculations carried out, the Z_{aval} and C parameters, therefore, do not influence the results: the sensitivity analysis shows that the variability of the Strickler coefficient has a predominant impact on the variability of the results at P67.200. Moreover, that this cross-section profile is far from the hydroelectric power station and the Villebois, it means that the hydraulic parameters involved in those model's profile do not contribute considerably to the water level at P67.200.

The uncertainty range of the Strickler coefficient is the most difficult to make because it depends on the flow, and the calibration data from which this evaluation was made are data for common operating flow rates. A sensitivity study to this range of variation has shown that uncertainty of +/- 10pt does not imply the same conclusions on the behavior of the model. When the amplitude of the variability of K increases, this parameter then becomes preponderant and the overall margin of error will be greater.

7 General conclusion

An improvement in the knowledge of the uncertain parameters would therefore make it possible to reduce the amplitude of their variability and thus reduce the amplitude of the confidence interval to 90% of the results.

A larger number of water surface profiles measurements carried out for the widest possible range of flow rates would improve the estimation of the variability, mainly the Strickler coefficient.

The Calibration methods that are more objective than those currently used would make it possible to find the most representative coefficients sets of all the measurements, and thus reduce the amplitude of the variability, this is why the first part of this work will be a subject for further research and development to enhance metamodel performance and prediction accuracy by training them on the more complex hydraulic model.

[32] It means the dam will ensure Its nominal functorialities without any problem.

8 Citations :

[1] Pappenberger, Beven, Horritt et Blazkova, «Uncertainty in the calibration of effective roughness parameters in HEC-RAS using inundation and downstream level observations,» *Journal of Hydrology,* pp. 302(1-4), 46-69, 2005.

[2] Sellin, Ervine et Willetts, «Behaviour of meandering two-stage channels,» *Proceedings of the Institution of Civil Engineers-Water Maritime and Energy,* pp. 101(2), 99-111, 1993.

[3] [Online]. Available: https://www.marinwatersheds.org/.

[4] [En ligne]. Available: http://www.opentelemac.org/index.php/modules-list/164-sysiphe-sediment-transport-and-bed-evolution.

[5] . P. . R. Wormleaton. et M. Karmegam, «Parameter optimization in flood routing,» *Journal of hydraulic engineering,* pp. 1799-1814, 1984.

[6] J. K. Lenstra et A. R. Kan, «Some simple applications of the travelling salesman problem,» *Journal of the Operational Research Society,* pp. 717-733, 1975.

[7] T. Rakić, . I. Kasagić-Vujanović, M. Jovanović, B. Jančić-Stojanović et D. Ivanović, «Comparison of full factorial design, central composite design, and box-behnken design in chromatographic method development for the determination of fluconazole and its impurities,» *Analytical Letters,* pp. 47(8), 1334-1347., 2014.

[8] C. Younhee, S. Doosam, Y. Sungmin et K. Junemo, «Comparison of Factorial and Latin Hypercube Sampling,» *energies,* 2021.

[9] [En ligne]. Available: https://xgboost.readthedocs.io/en/latest/.

[10] A. Saltelli, M. Ratto, S. Tarantola et F. Campolongo, «Sensitivity analysis practices: Strategies for model-based inference,» *Sensitivity analysis practices: Strategies for model-based inference. Reliability Engineering & System Safety,* pp. 91(10-11), 1109-1125, 2000.

[11] C. GOEURY, T. DAVID et Riadh, «Uncertainty Quantification on a real case with TELEMAC-2D,» *Proceedings of the XXII TELEMAC-MASCARET Technical User Conference,* pp. p. 44-51, 2015.

[12] T. HOMMA et A. SALTELLI, «Use of Sobol's quasirandom sequence generator for integration of modified uncertainty importance measure,» *Journal of Nuclear Science and Technology,* pp. vol. 32, no 11, p. 1164-1173., 1995.

[13] Q. Ren, M. Li et S. Han, «ectonic discrimination of olivine in basalt using data mining techniques based on major elements: a comparative study from multiple perspectives,» *Big Earth Data,* pp. 3(1), 8-25, 2019.

[14] C. Goeury, T. David, R. Ata, S. Boyaval, Y. Audouin, N. Goutal et R. Barate, «Uncertainty Quantification on a real case with TELEMAC-2D,» *Technical User Conference,* pp. 15-16, 2047 (pp. 44-51)., 2015.

[15] badre, ok, loit : 456456 , 20202 .

[16]

[17] D. Lilas, S. Proust, A. Paquier et N. Goutal, «Analyse de la pertinence du calage du coefficient de Manning pour des crues faiblement débordantes,» chez *Simhydro 2010*, 2010, June.

[18] J. Qi, Z. Xu, Y. Xue et Z. Wen, «An Efficient Method for Min-dist Location Selection».

[19] A. Visse, F.-X. Cierco , L. Duron, Y. Gressier et P.-L. Rothé, «Validation of a semi-automatically calibrated 1-D open-channel model against experimental data with changes in channel geometry,» chez *SimHydro 2019*, Sophia Antipolis , 2019.

[20] «Cross-validation: evaluating estimator performance,» [En ligne]. Available: https://scikit-learn.org/.

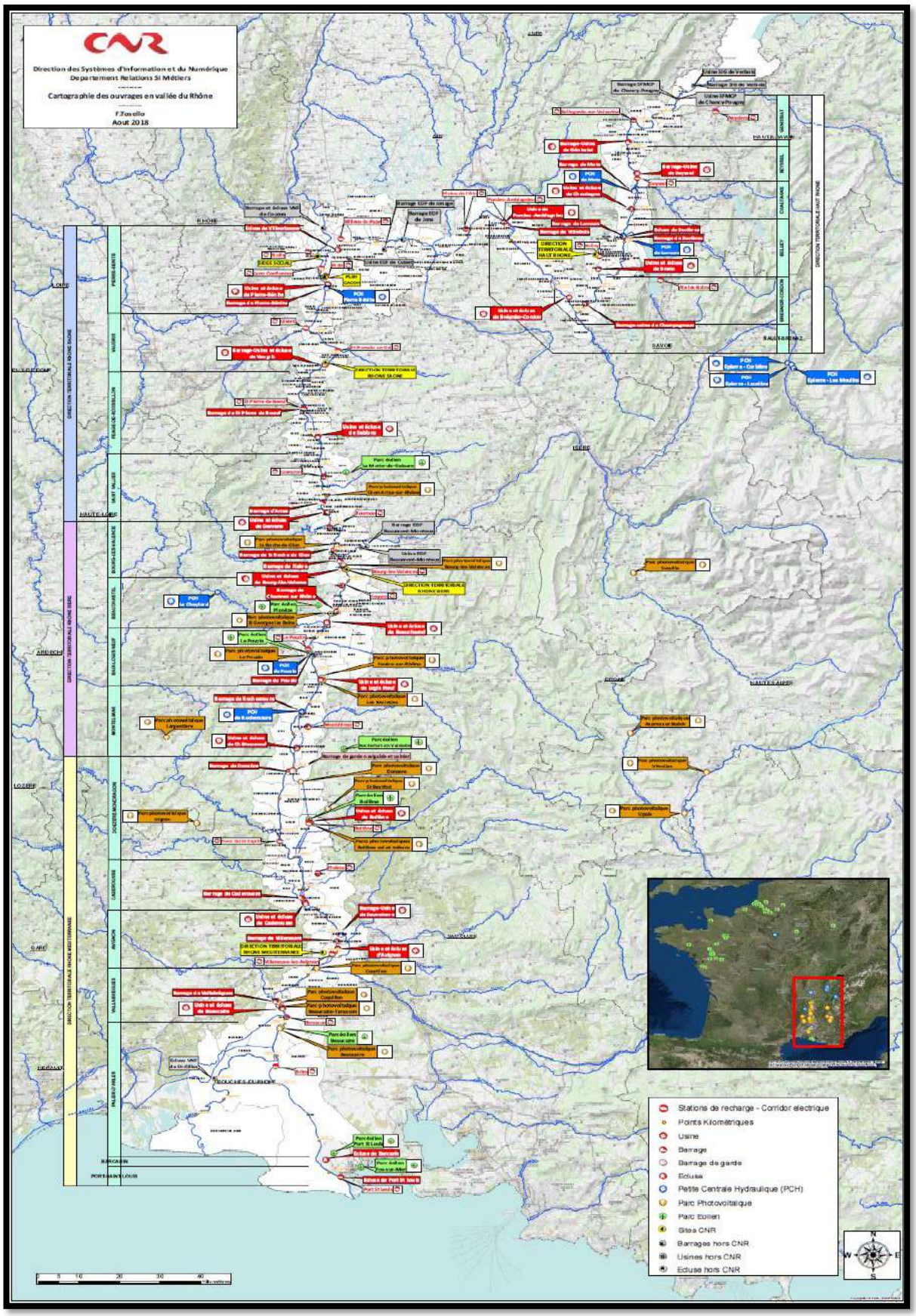
[21] «http://www.opentelemac.org/index.php/modules-list/164-sysiphe-sediment-transport-and-bed-evolution,» [En ligne].

[22] Y. Jung et V. Merwade, «Estimation of uncertainty propagation in flood inundation mapping using a 1-D hydraulic model,» *Hydrological Processes,* pp. 29(4), 624-640, 2015.

[23] C. GOEURY et T. DAVID, «Uncertainty Quantification on a real case with TELEMAC-2D,» *TELEMAC-MASCARET Technical User Conference,* pp. p. 44-51, 2015.

[24] C. GOEURY, Y. AUDOUIN et F. ZAOUI, «Interoperability applications of TELEMAC-MASCARET System,» *TELEMAC-MASCARET User Conference,* pp. p. 57-64, 2017.

[25] «persalys.github.io,» [En ligne]. Available: https://persalys.github.io/persalys/latest/user_manual/graphical_interface/probabilistic_analysis/user_manual_probabilistic_analysis.html.



Appendix 1: Hydrotechnical works on Rhône's valley. By Mr. F. TOSELLO

```
<Scenarios>
  <Scenario Nom="Sc_CE2016_PR1_Juin_2016">
    <Type>Crue10</Type>
    <IsActive>true</IsActive>
    <Commentaire>sans br15 et sans canal d&amp;apos;amené avec nouveaux K,S
    <AuteurCreation>LAPLACE</AuteurCreation>
    <DateCreation>2016-12-14T14:21:17.374</DateCreation>
    <AuteurDerniereModif>BENEFICE_G</AuteurDerniereModif>
    <DateDerniereModif>2020-06-19T13:57:26.641</DateDerniereModif>
    <Scenario-FichEtudes>
      <OCAL NomRef="CE2016_PR1_Juin_2016.ocal.xml"/>
      <ORES NomRef="CE2016_PR1_Juin_2016.ores.xml"/>
      <PCAL NomRef="CE2016_PR1_Juin_2016.pcal.xml"/>
      <DCLM NomRef="CE2016_PR1_Juin_2016.dclm.xml"/>
      <DLHY NomRef="CE2016_PR1_Juin_2016.dlhy.xml"/>
    </Scenario-FichEtudes>
    <Scenario-Modeles>
      <Scenario-Modele NomRef="Mo_CE2016_PR1_Juin_2016"/>
    </Scenario-Modeles>
    <Runs>
      <Run Nom="Iter0">
        <Commentaire></Commentaire>
        <AuteurCreation>ALLOUL</AuteurCreation>
        <DateCreation>2021-06-26T17:22:48.000</DateCreation>
        <AuteurDerniereModif>ALLOUL</AuteurDerniereModif>
        <DateDerniereModif>2021-06-26T17:22:48.000</DateDerniereModif>
      </Run>
      <Run Nom="Iter1">
        <Commentaire></Commentaire>
        <AuteurCreation>ALLOUL</AuteurCreation>
        <DateCreation>2021-06-26T17:22:48.000</DateCreation>
        <AuteurDerniereModif>ALLOUL</AuteurDerniereModif>
        <DateDerniereModif>2021-06-26T17:22:48.000</DateDerniereModif>
      </Run>
      <Run Nom="Iter2">
        <Commentaire></Commentaire>
        <AuteurCreation>ALLOUL</AuteurCreation>
        <DateCreation>2021-06-26T17:22:49.000</DateCreation>
        <AuteurDerniereModif>ALLOUL</AuteurDerniereModif>
        <DateDerniereModif>2021-06-26T17:22:49.000</DateDerniereModif>
      </Run>
      <Run Nom="Iter3">
        <Commentaire></Commentaire>
        <AuteurCreation>ALLOUL</AuteurCreation>
        <DateCreation>2021-06-26T17:22:50.000</DateCreation>
```

Appendix 2: *etu.xml file of the CE2016 model

```
<TypeCalage>
  <Donnees>Permanent</Donnees>
  <Critere>ErreurQuadratique</Critere>
</TypeCalage>
<Calage>
  <Type>TestRepetabilite</Type>
  <Algorithme>Recuit</Algorithme>
  <Seul>
    <NombreIteration>1000</NombreIteration>
  </Seul>
  <TestRepetabilite>
    <NombreIterationTir>1000</NombreIterationTir>
    <NombreTir>10</NombreTir>
  </TestRepetabilite>
  <ValidationCroisee>
    <NombreIteration>1</NombreIteration>
    <PonderationApprentissage>0.0</PonderationApprentissage>
    <PonderationValidation>1.0</PonderationValidation>
  </ValidationCroisee>
</Calage>
<DonneesCampagne>
  <LoisCalculsPermanents>
    <LoiCalculPermanent CalculRef="Cc_P009_LE20160426" LoiRef="LoiSectionsZ_20160426" Ponderation="1"/>
    <LoiCalculPermanent CalculRef="Cc_P012_LE20160202" LoiRef="LoiSectionsZ_20160202" Ponderation="1"/>
  </LoisCalculsPermanents>
  <LoisCalculsTransitoires/>
  <LoisStrickler>
    <LoiStrickler LoiRef="Fk_RET1Maj" Min="5" Ini="15" Max="40"/>
    <LoiStrickler LoiRef="Fk_RET1Min" Min="20" Ini="30" Max="60"/>
    <LoiStrickler LoiRef="Fk_RET2Maj" Min="5" Ini="15" Max="40"/>
    <LoiStrickler LoiRef="Fk_RET2Min" Min="20" Ini="30" Max="60"/>
    <LoiStrickler LoiRef="Fk_RET3Maj" Min="5" Ini="15" Max="40"/>
    <LoiStrickler LoiRef="Fk_RET3Min" Min="20" Ini="30" Max="60"/>
    <LoiStrickler LoiRef="Fk_RET4Maj" Min="5" Ini="15" Max="40"/>
    <LoiStrickler LoiRef="Fk_RET4Min" Min="20" Ini="30" Max="60"/>
    <LoiStrickler LoiRef="Fk_RET5Maj" Min="5" Ini="15" Max="40"/>
    <LoiStrickler LoiRef="Fk_RET5Min" Min="20" Ini="30" Max="60"/>
    <LoiStrickler LoiRef="Fk_CACE1Maj" Min="5" Ini="15" Max="40"/>
    <LoiStrickler LoiRef="Fk_CACE1Min" Min="20" Ini="30" Max="60"/>
  </LoisStrickler>
  <ListeCalculs>
    <ParamCalc CalculRef="Cc_P009_LE20160426" EMHRef="Nd_RET151.520" DeltaQ="0.0"/>
    <ParamCalc CalculRef="Cc_P009_LE20160426" EMHRef="Nd_FIER8" DeltaQ="0.0"/>
    <ParamCalc CalculRef="Cc_P009_LE20160426" EMHRef="Nd_BARCE" DeltaQ="0.0"/>
    <ParamCalc CalculRef="Cc_P012_LE20160202" EMHRef="Nd_RET151.520" DeltaQ="0.0"/>
    <ParamCalc CalculRef="Cc_P012_LE20160202" EMHRef="Nd_FIER8" DeltaQ="0.0"/>
```

The flexibility (ΔQ) on node: "Nd_RET151.520" of the water surface profile "Cc_P009_LE20160426".

Appendix 3: *aoc.xml file of the CE2016 model

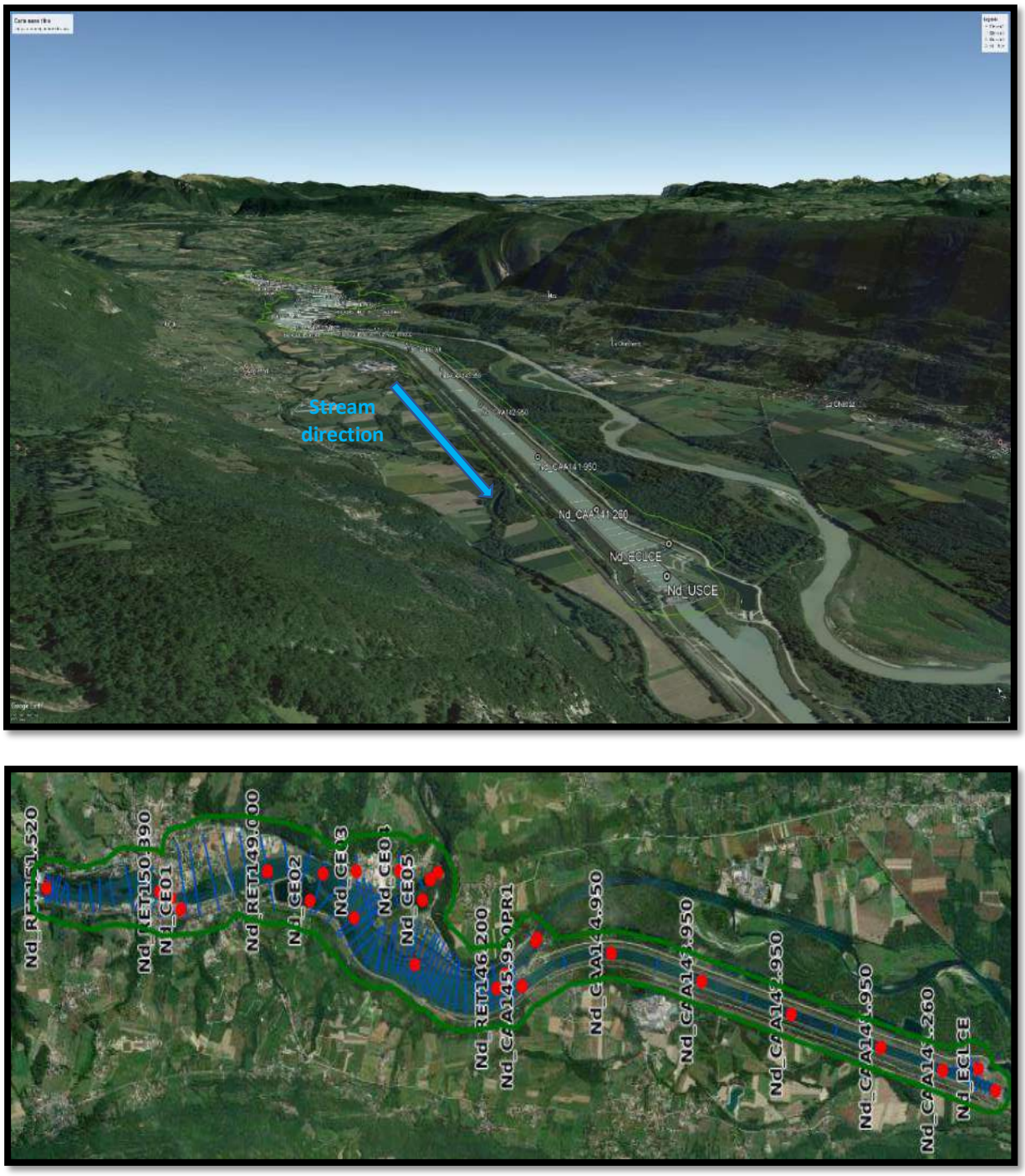
```
  <EchelleSection PK="151.300" SectionRef="St_RET151.300"/>
  <EchelleSection PK="151.000" SectionRef="St_RET150.990"/>
  <EchelleSection PK="150.600" SectionRef="St_RET150.530"/>
  <EchelleSection PK="150.300" SectionRef="St_RET150.300"/>
  <EchelleSection PK="150.150" SectionRef="St_RET150.150"/>
  <EchelleSection PK="150.000" SectionRef="St_RET150.000"/>
  <EchelleSection PK="149.750" SectionRef="St_RET149.750"/>
  <EchelleSection PK="149.000" SectionRef="St_RET149.000"/>
  <EchelleSection PK="148.665" SectionRef="St_RET148.700"/>
  <EchelleSection PK="148.500" SectionRef="St_RET148.500"/>
  <EchelleSection PK="148.165" SectionRef="St_RET148.200"/>
  <EchelleSection PK="147.900" SectionRef="St_RET147.850"/>
  <EchelleSection PK="147.500" SectionRef="St_RET147.500"/>
  <EchelleSection PK="146.100" SectionRef="St_CAA146.130"/>
  <EchelleSection PK="145.900" SectionRef="St_CAA145.950"/>
</EchellesSections>
<LoisLHPT>
  <LoiTF Nom="LoiSectionsZ_20160202" Type="LoiSectionsZ">
    <Commentaire></Commentaire>
    <EvolutionTF>
      <PointTF>151.300 252.58</PointTF>
      <PointTF>151.000 252.4</PointTF>
      <PointTF>150.600 252.32</PointTF>
      <PointTF>150.300 252.24</PointTF>
      <PointTF>150.150 252.13</PointTF>
      <PointTF>150.000 252.07</PointTF>
      <PointTF>149.750 252.03</PointTF>
      <PointTF>149.000 251.98</PointTF>
      <PointTF>148.665 251.93</PointTF>
      <PointTF>148.500 251.93</PointTF>
      <PointTF>148.165 251.92</PointTF>
      <PointTF>147.900 251.93</PointTF>
      <PointTF>147.500 251.91</PointTF>
      <PointTF>146.100 251.86</PointTF>
      <PointTF>145.900 251.87</PointTF>
    </EvolutionTF>
  </LoiTF>
  <LoiTF Nom="LoiSectionsZ_20160426" Type="LoiSectionsZ">
    <Commentaire></Commentaire>
    <EvolutionTF>
      <PointTF>151.300 252.4</PointTF>
      <PointTF>151.000 252.3</PointTF>
      <PointTF>150.600 252.23</PointTF>
      <PointTF>150.300 252.16</PointTF>
      <PointTF>150.150 252.09</PointTF>
```

Appendix 4: *lhpt.xml file of the CE2016 model

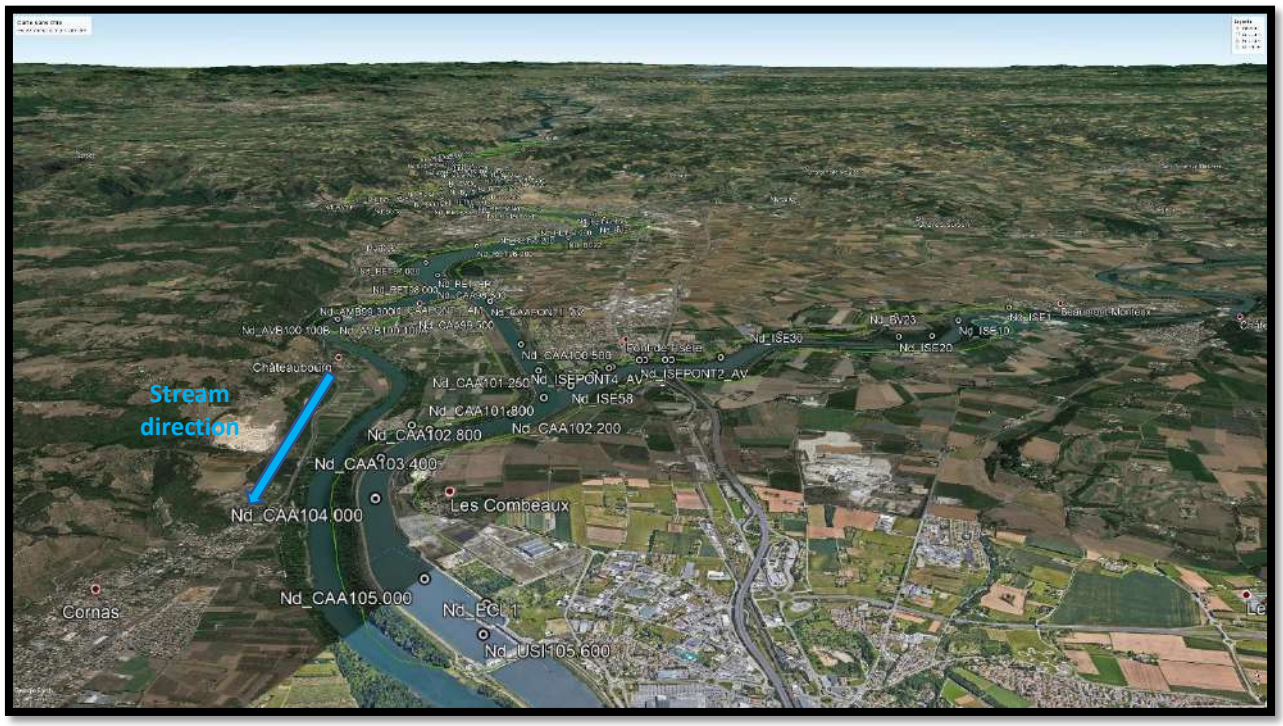
```
<Commentaire>Modèle vierge</Commentaire>
<CalcPseudoPerm Nom="Cc_P003">
  <Commentaire></Commentaire>
  <CalcPseudoPermNoeudQapp NomRef="Nd_RET151.520">
    <IsActive>true</IsActive>
    <Qapp>250.0</Qapp>
  </CalcPseudoPermNoeudQapp>
  <CalcPseudoPermNoeudQapp NomRef="Nd_FIER8">
    <IsActive>true</IsActive>
    <Qapp>15.0</Qapp>
  </CalcPseudoPermNoeudQapp>
  <CalcPseudoPermNoeudQapp NomRef="Nd_BARCE">
    <IsActive>true</IsActive>
    <Qapp>-70.0</Qapp>
  </CalcPseudoPermNoeudQapp>
  <CalcPseudoPermNoeudNiveauContinuZimp NomRef="Nd_CAA145.950PR1">
    <IsActive>true</IsActive>
    <Zimp>251.99867</Zimp>
  </CalcPseudoPermNoeudNiveauContinuZimp>
</CalcPseudoPerm>
<CalcPseudoPerm Nom="Cc_P004">
  <Commentaire></Commentaire>
  <CalcPseudoPermNoeudQapp NomRef="Nd_RET151.520">
    <IsActive>true</IsActive>
    <Qapp>300.0</Qapp>
  </CalcPseudoPermNoeudQapp>
  <CalcPseudoPermNoeudQapp NomRef="Nd_FIER8">
    <IsActive>true</IsActive>
    <Qapp>15.0</Qapp>
  </CalcPseudoPermNoeudQapp>
  <CalcPseudoPermNoeudQapp NomRef="Nd_BARCE">
    <IsActive>true</IsActive>
    <Qapp>-70.0</Qapp>
  </CalcPseudoPermNoeudQapp>
  <CalcPseudoPermNoeudNiveauContinuZimp NomRef="Nd_CAA145.950PR1">
    <IsActive>true</IsActive>
    <Zimp>251.99746</Zimp>
  </CalcPseudoPermNoeudNiveauContinuZimp>
</CalcPseudoPerm>
<CalcPseudoPerm Nom="Cc_P005">
  <Commentaire></Commentaire>
  <CalcPseudoPermNoeudQapp NomRef="Nd_RET151.520">
    <IsActive>true</IsActive>
    <Qapp>307.0</Qapp>
  </CalcPseudoPermNoeudQapp>
  <CalcPseudoPermNoeudQapp NomRef="Nd_FIER8">
```

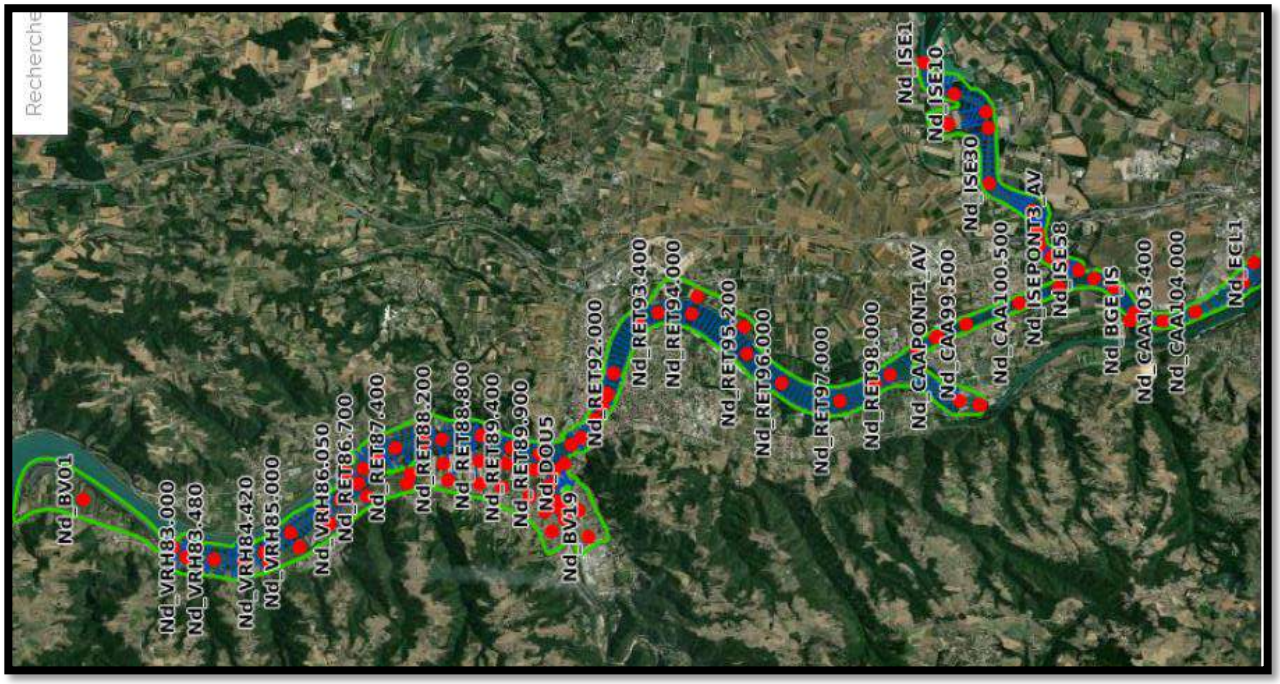
The discharge value (Q) on node: "Nd_RET151.520" of the water surface profile "Cc_P003".

Appendix 5: *dclm. file of the CE2016 model

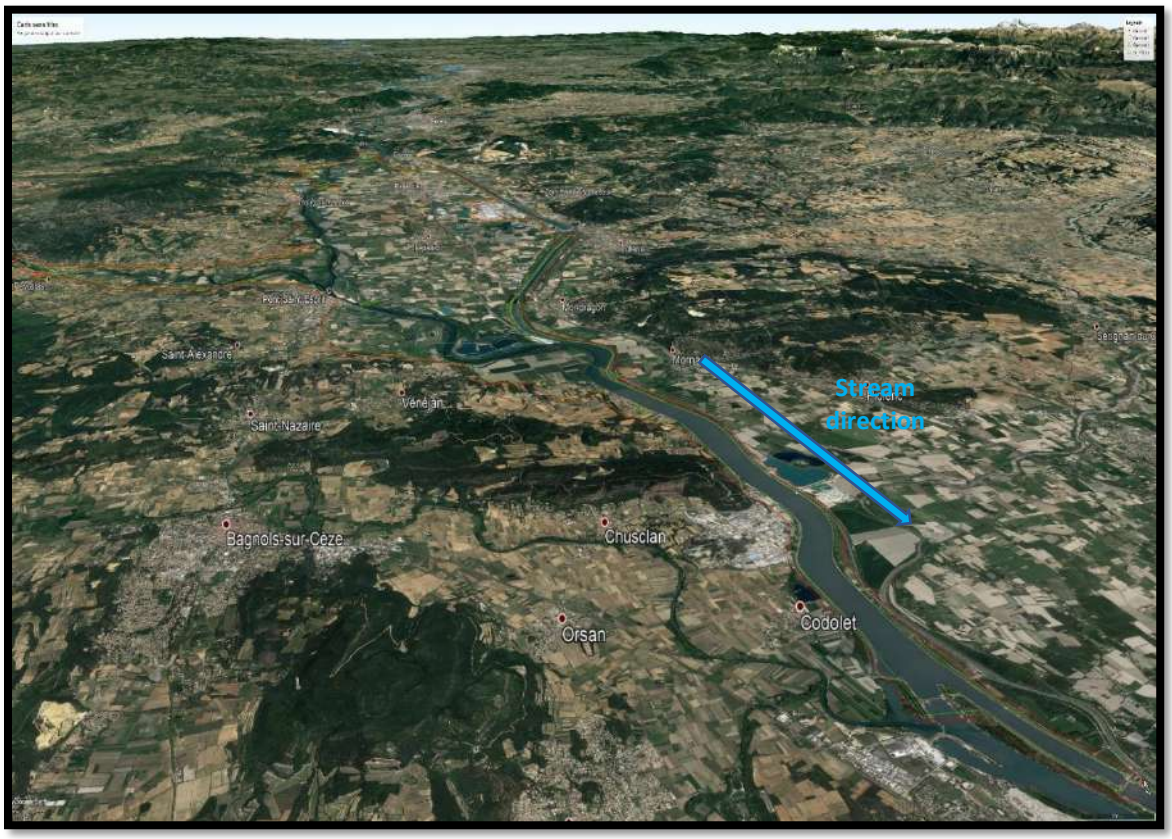


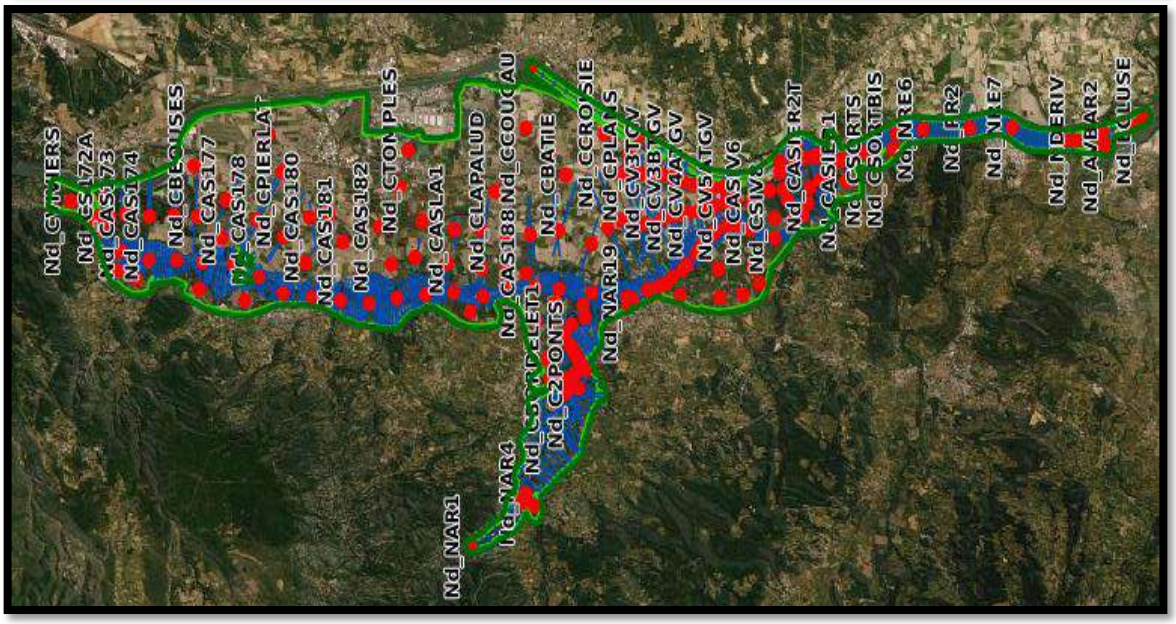
Appendix 6: The Chautagne hydraulic model 2016 (CA2016)



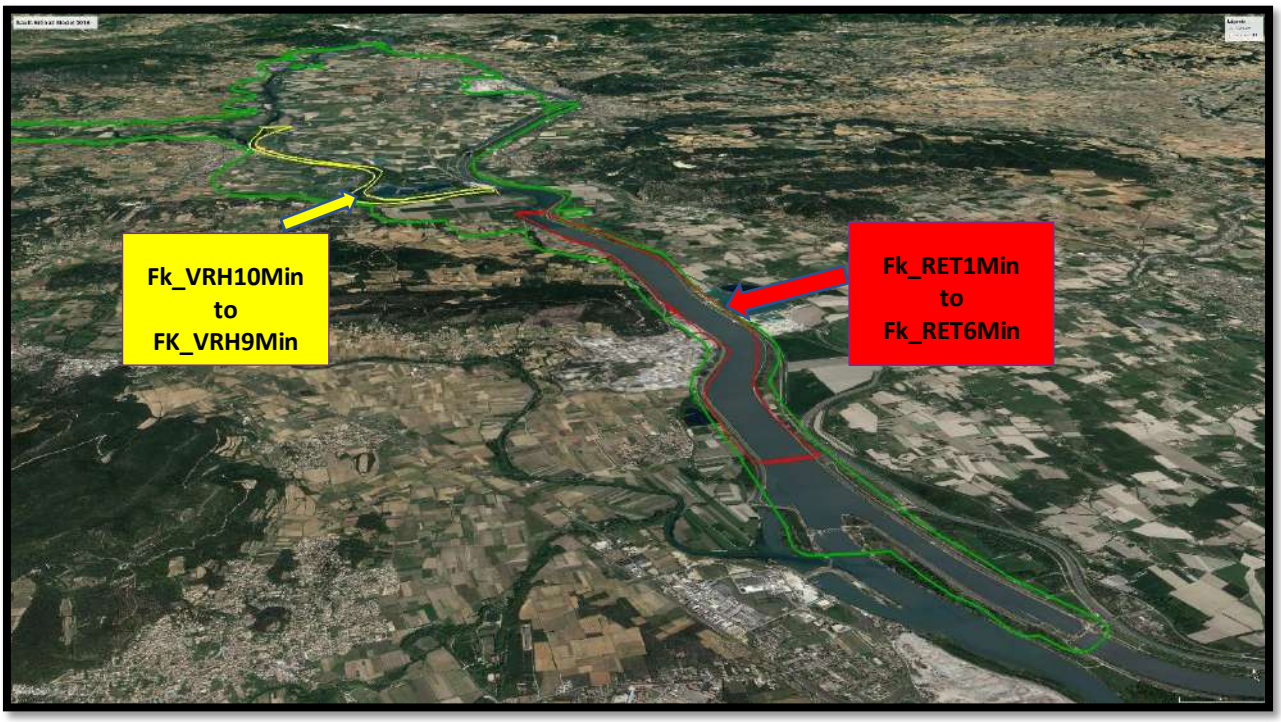


Appendix 7: The Bourg-Lès-Valence hydraulic model 2016 (BV2016)

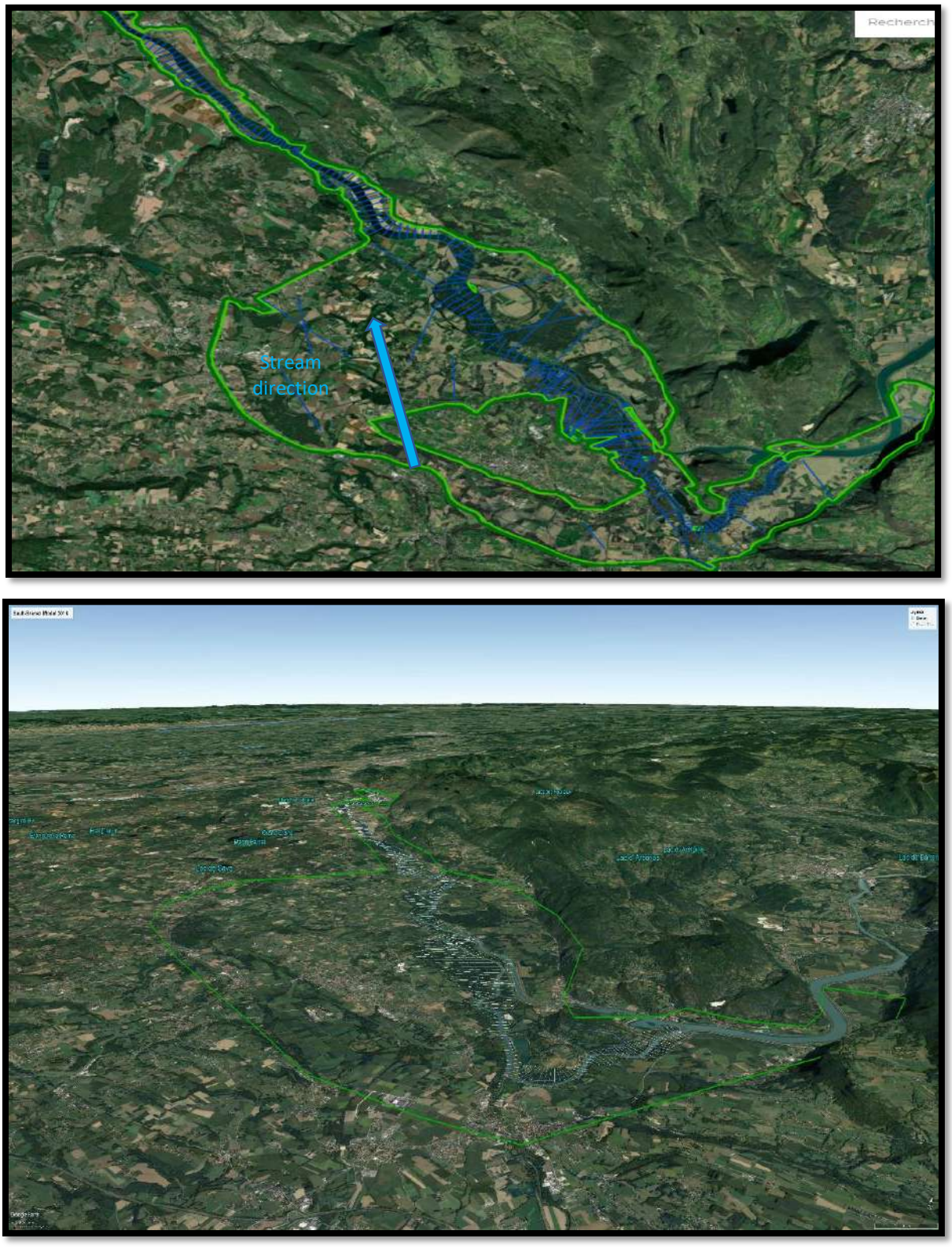




Appendix 8: The Caderousse hydraulic model (CA2020)



Appendix 8α: CA2020 model Strickler zones precision



Appendix 9: The Sault-Brénaz hydraulic model (SB2013)

Fk_RET4Min	Fk_RET5Maj	Fk_RET5Min	Fk_CACE1Maj	Fk_CACE1Min	Cc_P009_LE20160426/151.300	Cc_P009_LE20160426/151.000
30.15061898	19.28350233	54.37089036	38.72447641	30.83026601	252.1558225	252.1012932
59.27498757	26.33979619	23.23004451	33.85231104	26.63242822	252.2633481	252.1462898

Appendix 10: Example of Two rows of 4 calibration parameters presenting the DOE construction, and 2 water levels at two different sections (151.300 and 151) for the same water surface profiles (Cc_P009_LE20160425) which are the simulation code result. The whole table is the dataset.

```
def prepare_inputs(x, nb_lois_frottement, input_design=None):
    """
    Processes input data
    :type nb_lois_frottement: int
    :param nb_lois_frottement: Number of Strikcler coefficients
    :param x: inputs
    :type x: np.array, shape(num_samples, num_input)
    :param input_design: list of input power transformations (optional)
    :type input_design: list
    :return: processed input data
    :rtype: np.array, shape(num_samples, num_input*len(input_design))
    """
    if input_design is None:
        input_design = [[-1, -3 / 5], [1, 3 / 5]]

    l_concat = []

    if len(x.shape) > 1:  # Multiple rows

        if x.shape[1] > nb_lois_frottement:  # With Q_factors and physical consideration

            striklers = np.concatenate(
                [np.power(x[:, :nb_lois_frottement], input_design[0][i]) for i in range(len(input_design[0]))]
                , axis=1)
            qfact = np.concatenate(
                [np.power(x[:, -(x.shape[1] - nb_lois_frottement):], input_design[1][i]) for i in
                 range(len(input_design[1]))]
                , axis=1)
            inputs = np.concatenate([striklers, qfact], axis=1)

        else:  # Without Q_factors
            if len(input_design) > 1:
                for i in input_design[0]:
                    l_concat.append(x ** i)
            else:
                for i in input_design:
                    l_concat.append(x ** i)

            if len(x.shape) > 1:  # With rows lignes
                inputs = np.concatenate(l_concat, axis=1)
            else:
                inputs = np.concatenate(l_concat)

    else:  # One numpy vector
        if x.shape[0] > nb_lois_frottement:  # With Q_factors

            striklers = np.concatenate(
                [np.power(x[:nb_lois_frottement], input_design[0][i]) for i in range(len(input_design[0]))]
```

Appendix 11 : The input power. Different powers were programmed to be attributed to the inputs (The list which contains the powers values are called input design in the code).

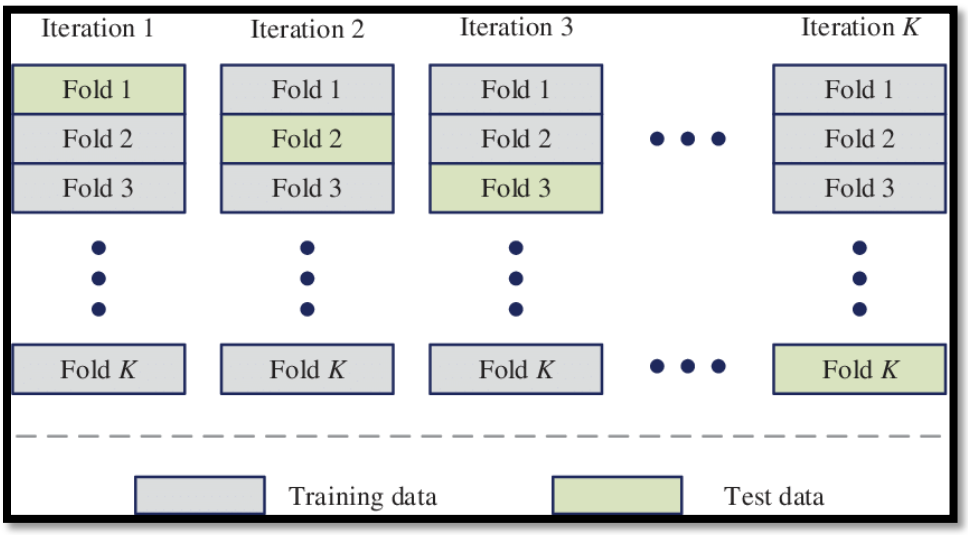
```python
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import Matern, ConstantKernel, RBF, RationalQuadratic, DotProduct, WhiteKernel, \
    ExpSineSquared


def gaussian_process_regressor(x, y):
    """
    Gaussian Process Regressor metamodel

    :param x: matrix with inputs
    :param y: matrix with outputs
    :return: GPR metamodel
    :rtype: sklearn.gaussian_process.gpr.GaussianProcessRegressor
    """
    kernel = ConstantKernel() * DotProduct() + ConstantKernel() * RationalQuadratic() + WhiteKernel()
    gp = GaussianProcessRegressor(kernel=kernel, n_restarts_optimizer=10)
    gp.fit(x, y)
    print(gp.kernel_)

    return gp
```

Appendix 12: The Gaussian process algorithm built-on Scikit-learn library (Kernel 33 given by Yang et al., TUC 2020)

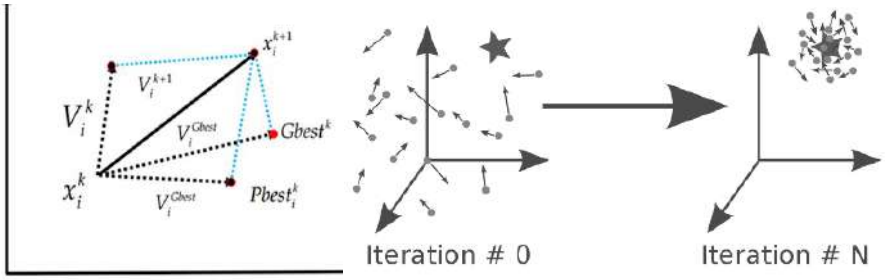
```python
import sklearn.multioutput
import openturns as ot


def ot_gpr(x, y):
    from otsklearn import Kriging

    covarianceModel = ot.MaternModel([1.0] * x.shape[1], 2.5)
    single_model = Kriging(kernel=covarianceModel, basis="Linear")
    model = sklearn.multioutput.MultiOutputRegressor(single_model)
    model.fit(x, y)
    return model
```

Appendix 13 : The Gaussian process built-on OpenTurns library

[33] There is no scientific proof for the kernel construction efficiency, it is a developer choice.



Appendix 14: K-fold cross-validation method illustration. [13]



At each iteration, particles are moved according to 3 components:

- Current velocity V_k,
- Its best solution P_i,
- Its best solution obtained in its vicinity P_g.

This gives the following equation of motion:

$$V_{k+1} = \omega V_k + b_1(P_i - 1) + b_2(P_g - X_k)$$

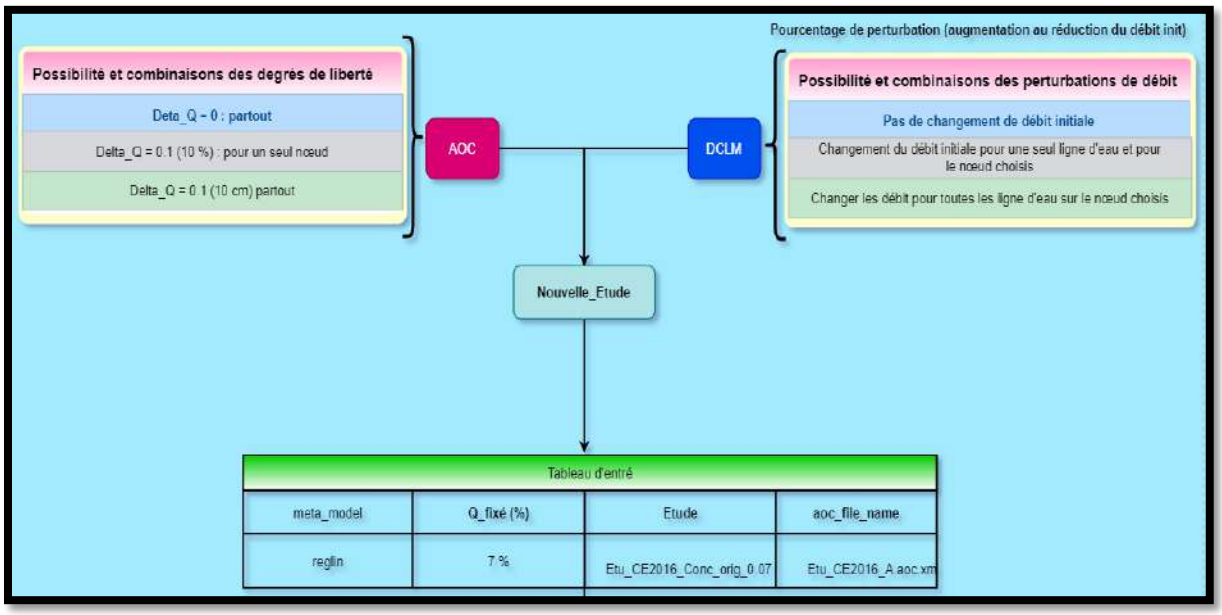
$$X_{k+1} = X_k + V_{k+1}$$

With ω an inertia parameter, b_1 taken randomly from $[0, \varphi_1]$, b_2 taken randomly in $[0, \varphi_2]$ and φ_1, φ_2 are two hyperparameters

Appendix 15: PSO algorithm illustration

	Cc_P_20010310	Cc_P_20021	Cc_P_20100	Cc_P_20150331	Cc_P_2016061	mean	robustness
rmse 0	0.052	0.068	0.037	0.075	0.056	0.058	1.461
rmse 1	0.042	0.076	0.039	0.079	0.053	0.058	1.852
rmse 2	0.043	0.067	0.035	0.075	0.051	0.054	1.651
rmse 3	0.045	0.078	0.039	0.079	0.054	0.059	1.869
rmse 4	0.056	0.076	0.041	0.075	0.059	0.061	1.467
rmse 5	0.053	0.066	0.041	0.076	0.059	0.059	1.325
rmse 6	0.046	0.083	0.039	0.079	0.058	0.061	1.937
rmse 7	0.056	0.068	0.039	0.078	0.062	0.061	1.444
rmse 8	0.047	0.064	0.037	0.075	0.056	0.056	1.459
rmse 9	0.054	0.070	0.037	0.076	0.058	0.059	1.532
rmse 10	0.050	0.063	0.036	0.077	0.057	0.057	1.493
rmse 11	0.050	0.060	0.036	0.075	0.058	0.056	1.430
rmse 12	0.053	0.058	0.039	0.080	0.066	0.059	1.543
rmse 13	0.042	0.077	0.040	0.078	0.053	0.058	1.856
rmse 14	0.053	0.070	0.039	0.074	0.056	0.058	1.404
rmse 15	0.045	0.068	0.038	0.077	0.055	0.057	1.615
rmse 16	0.042	0.062	0.037	0.079	0.057	0.055	1.651
rmse 17	0.049	0.057	0.036	0.077	0.061	0.056	1.522
rmse 18	0.052	0.059	0.038	0.076	0.059	0.057	1.390
repeatability	0.482	0.736	0.167	0.171	0.345		

Appendix 16 : Robustness and Repeatability computation table (BV2016)



Appendix 17: The different steps to add a second calibration parameter to the problem

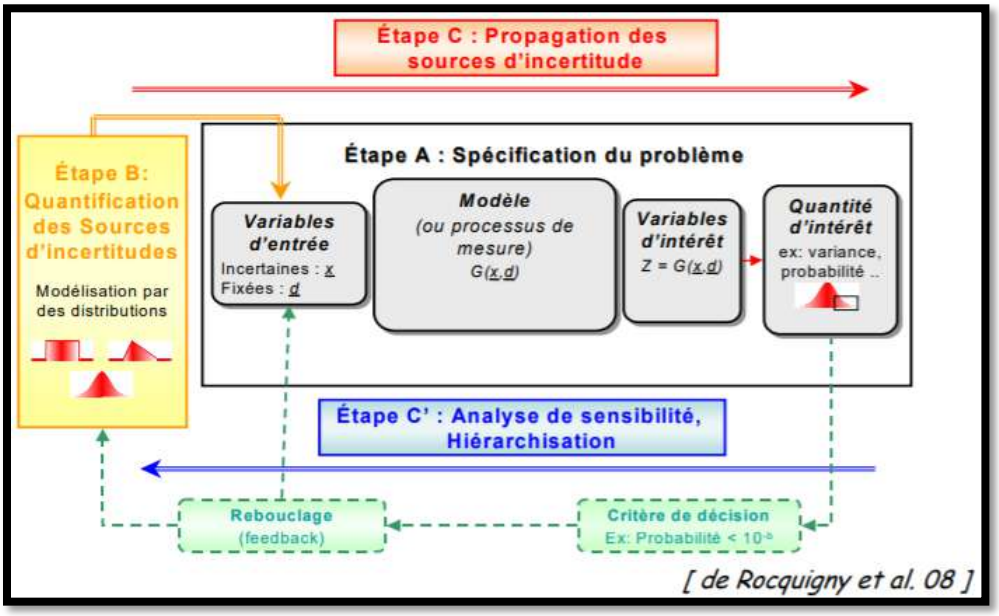
Predicted factor Q_{factor}	Learner	±7%		±5%		±2%	
		Dispersion	Prediction	Dispersion	Prediction	Dispersion	Prediction
Q Nd_RET151.520 of Cc_P012_LE20160202	Reglin	0	0.5	-0.5	1	-1	-1
	gpr	-0.5	-1	0	-1	-1	-1
	rdg	-1	1	1	-1	-1	-1
Q Nd_RET151.520 of Cc_P009_LE20160426	Reglin	1	-1	0	-1	-1	-1
	gpr	1	-1	-1	-1	-1	-1
	rdg	0	-1	0	-1	-1	-1

Appendix 18: Data categorization for "two perturbation" case predictions results

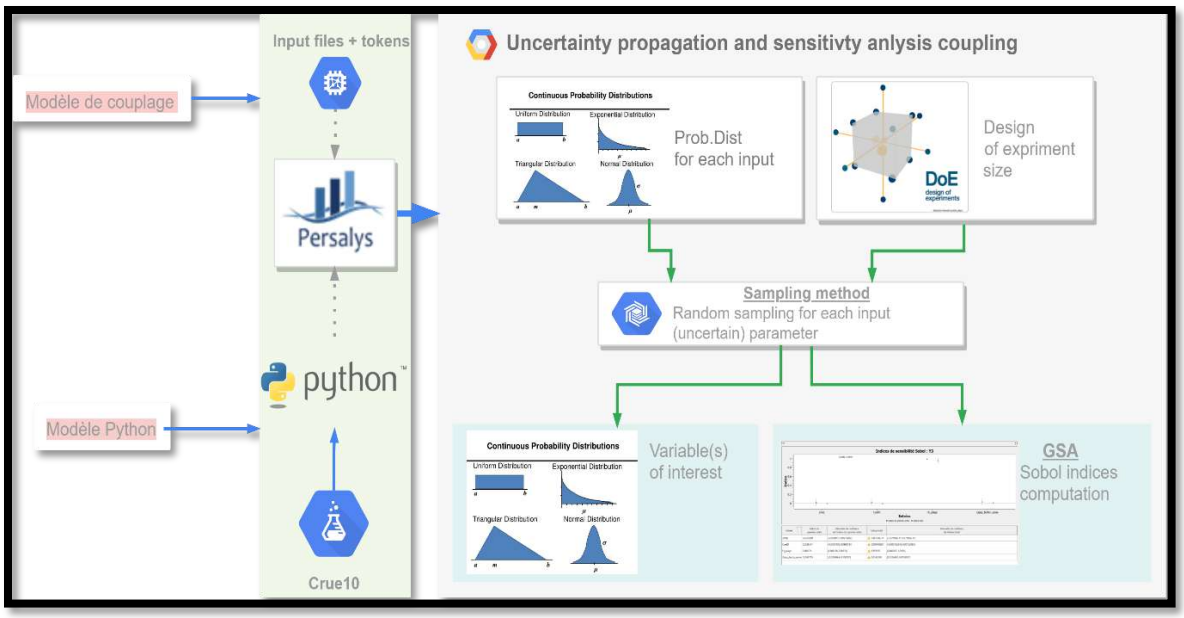
- Very good ➔ 1
- Relatively acceptable ➔ 0.5
- Good ➔ 0
- Worst ➔ -1

Input	Description	Unit	Probability distribution
Q	Maximal annual flowrate	m^3/s	Truncated Gumbel $\mathcal{G}(1013, 558)$ on $[500, 3000]$
K_s	Strickler coefficient	-	Truncated normal $\mathcal{N}(30, 8)$ on $[15, +\infty[$
Z_v	River downstream level	m	Triangular $\mathcal{T}(49, 50, 51)$
Z_m	River upstream level	m	Triangular $\mathcal{T}(54, 55, 56)$
H_d	Dyke height	m	Uniform $\mathcal{U}[7, 9]$
C_b	Bank level	m	Triangular $\mathcal{T}(55, 55.5, 56)$
L	Length of the river stretch	m	Triangular $\mathcal{T}(4990, 5000, 5010)$
B	River width	m	Triangular $\mathcal{T}(295, 300, 305)$

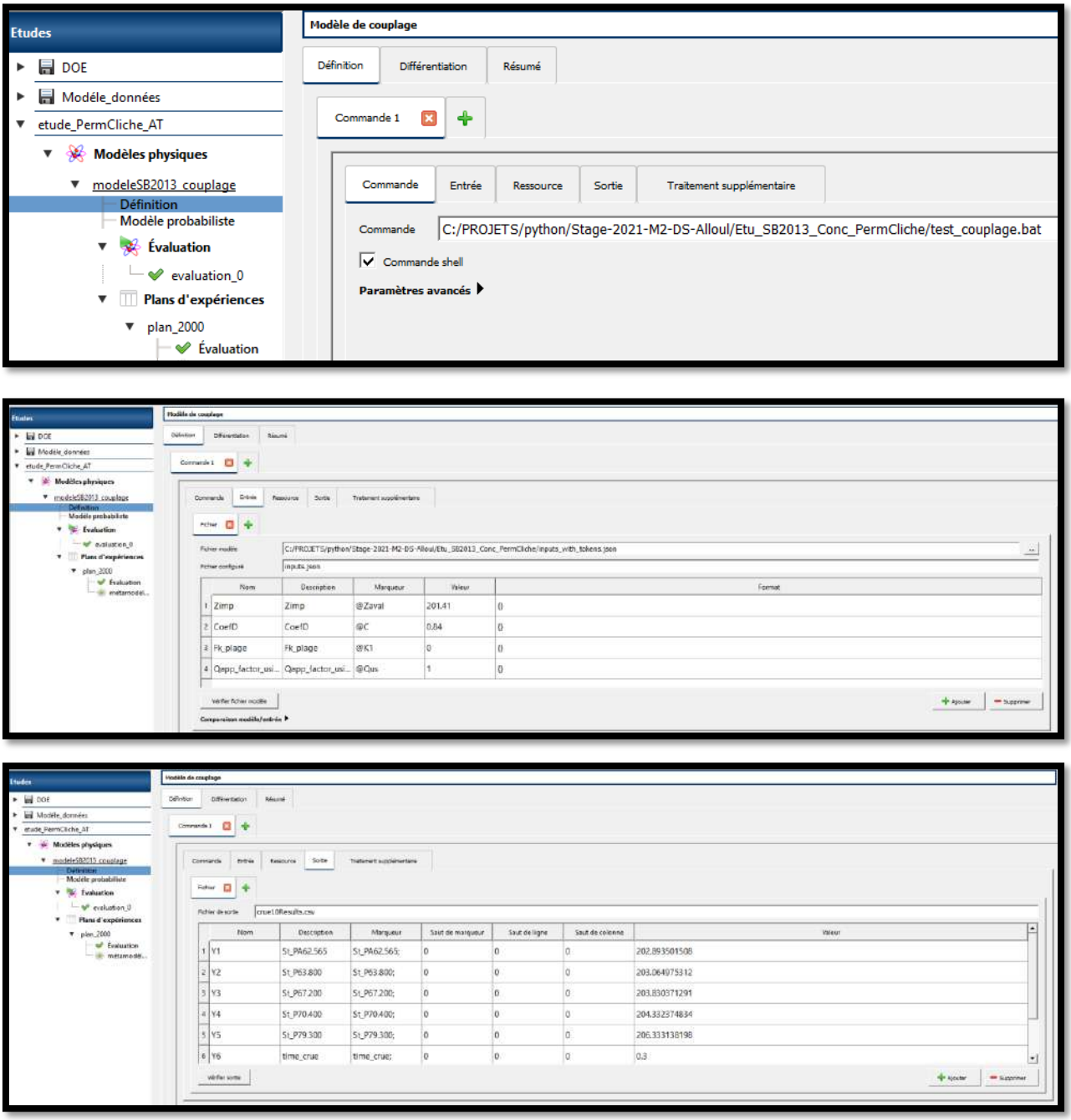
Appendix 19: Example of PDFs attribution for different input variables



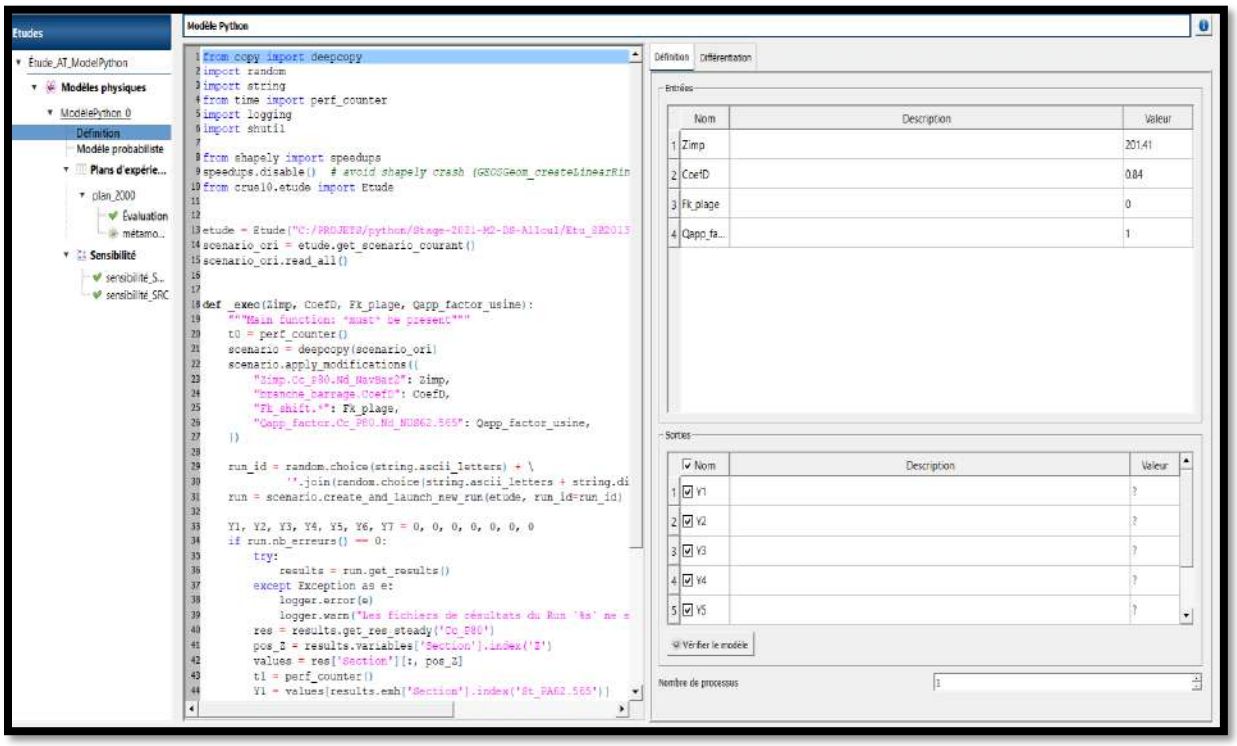
Appendix 20 : Uncertainty propagation study framework and steps [14]



Appendix 21: Steps for sensitivity analysis – PersalysCrue10 Coupling methods



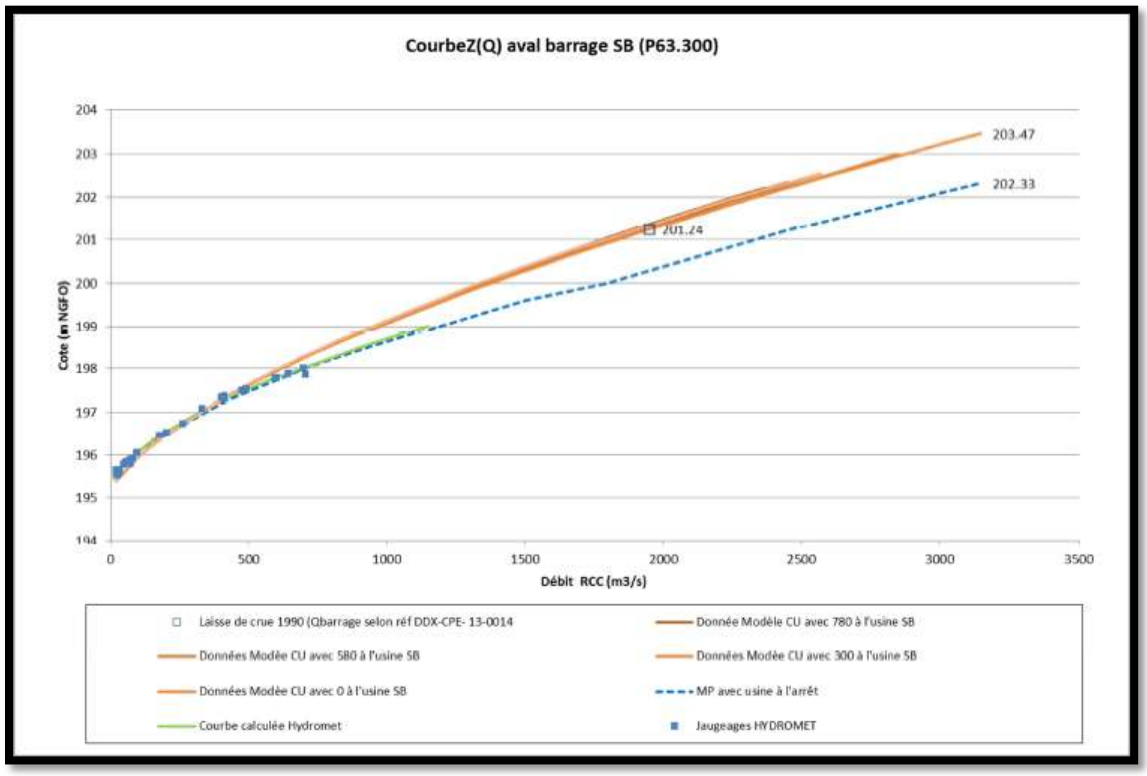
Appendix 22: **Modèle de couplage** -- Simulation code's path window

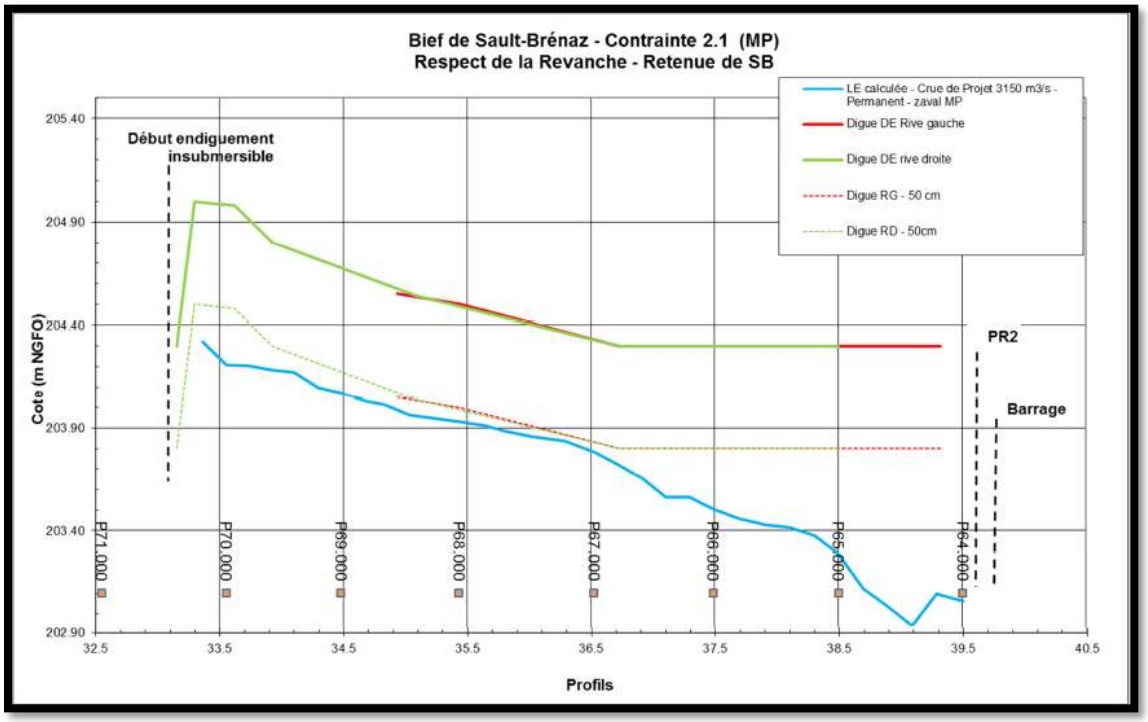


Appendix 23: **Modèle Python** -- Python code coupling with Persalys

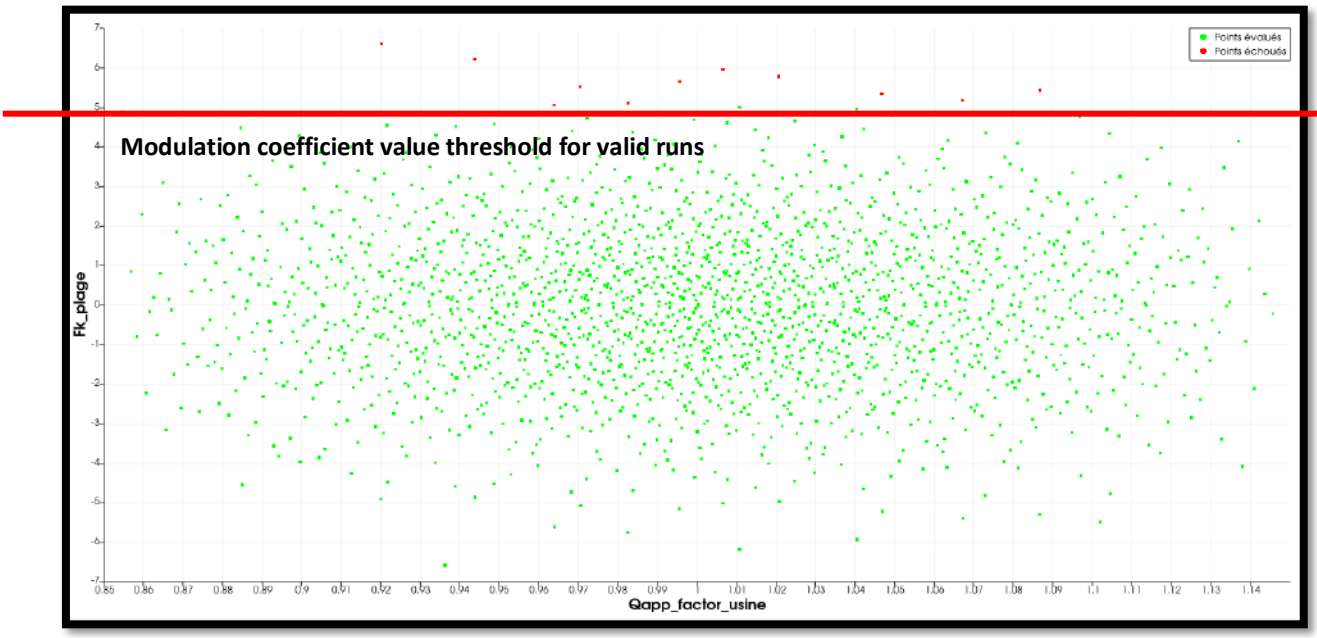
	Prométhée-Crue	Persalys (GUI)	OpenTurns (Python library)
Uncertainty propagation method		• Monte Carlo • Quasi-Monte Carlo • LHS	• Monte Carlo • Quasi-Monte Carlo • LHS
Problem dimensionality reduction (screening)	Morris Method	Morris Method	• OAT-based algorithms: One at the time • Morris Method
Sensitivity analysis	Fast method for Sobol indices	Sobol indices : a) Saltelli approach. b) PCE Regression-based : SRC indice	• Sobol indices (par PCE) • Sobol indices (par Saltelli)

Appendix 24 : Persalys Prométhée-Crue overview





Appendix 25: Available Villebois' dam downstream stage-discharge relationship



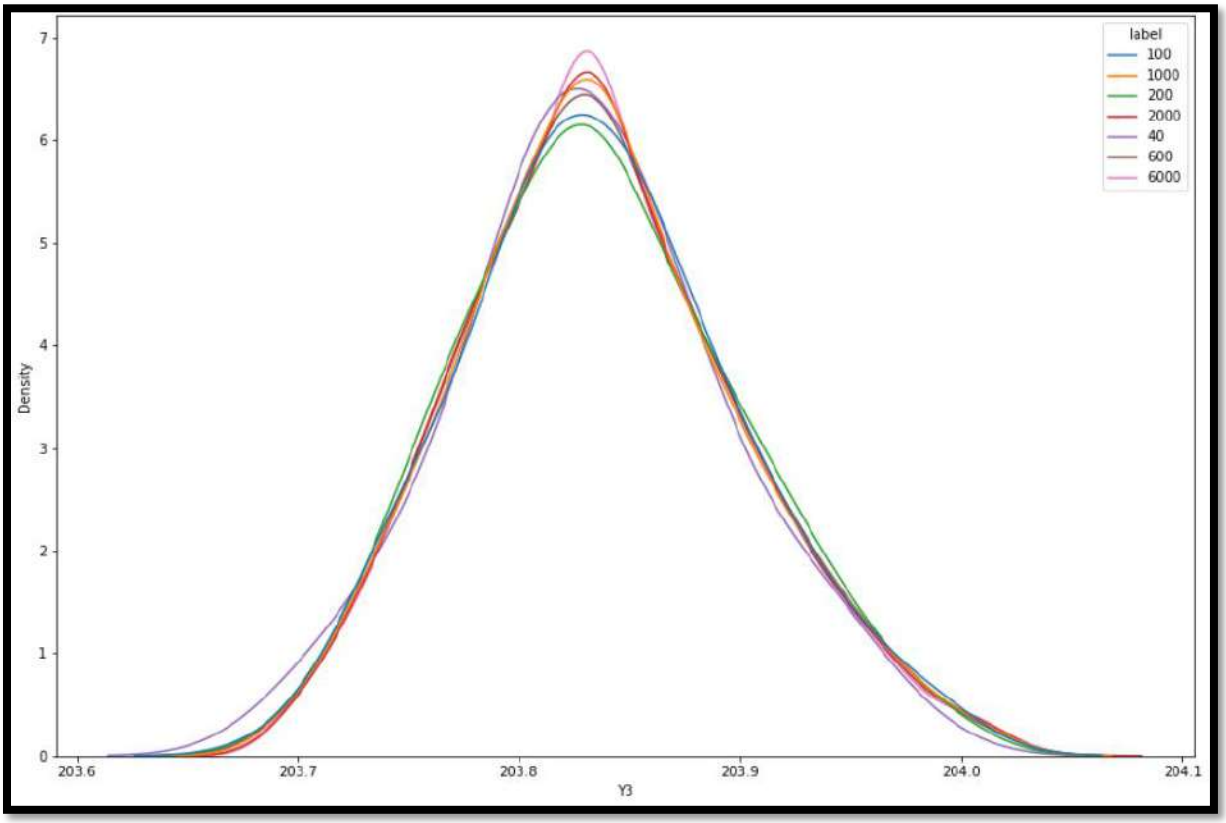
Appendix 26: Modulation coefficient values dispersion and truncation justification

Appendix 27: Profile P67.200. Model SB2013

Uncertain parameter / Input	Minimum sampling value	Maximum sampling value	Mean	Standard deviation	Nominal value	5% quantile (for Gaussian distribution)	95% quantile (for Gaussian distributions)
Qfactor	0.855	1.145	1.000	0.061	1.000	/	/
C	0.675	1.005	0.840	0.061	0.840	0.840	0.922
Zaval	201.023	201.822	0.840	0.125	201.400	201.410	201.616
K1	-4.947	4.947	-0.003	1.907	0.000	-0.001	3.176

Appendix 28 : Uncertain parameters probabilistic characteristics

Appendix 29: Different conducted studies with their DOEs sizes



Appendix 30: PDFs of Y3 (P67.200) by the simulation code Crue10 for different DOE sizes.

Sensitivity analysis using Sobol indices

This method deals with analyzing the influence the random vector $\boldsymbol{X} = (X^1, \ldots, X^{n_X})$ has on a random variable Y^k which is being studied for uncertainty. Here we attempt to evaluate the part of variance of Y^k due to the different components X^i.

The estimators for the mean of m_{Y^j} and the standard deviation σ of Y^k can be obtained from a first sample, as Sobol indices estimation requires two samples of the input variables : $(X^1, \ldots, X^{n_X})$, that is two sets of N vectors of dimension n_X $(x_{11}^{(1)}, \ldots, x_{1n_X})^{(1)}, \ldots, (x_{N1}^{(1)}, \ldots, x_{Nn_X}^{(1)})$ and $(x_{11}^{(2)}, \ldots, x_{1n_X})^{(2)}, \ldots, (x_{N1}^{(2)}, \ldots, x_{Nn_X}^{(2)})$

The estimation of sensitivity indices for first order consists in estimating the quantity

$$V_i = \mathrm{Var}\left[\mathbb{E}\left[Y^k|X_i\right]\right] = \mathbb{E}\left[\mathbb{E}\left[Y^k|X_i\right]^2\right] - \mathbb{E}\left[\mathbb{E}\left[Y^k|X_i\right]\right]^2 = U_i - \mathbb{E}\left[Y^k\right]^2$$

Sobol proposes to estimate the quantity $U_i = \mathbb{E}\left[\mathbb{E}\left[Y^k|X_i\right]^2\right]$ by swapping every variables in the two samples except the variable X_i between the two calls of the function:

$$\hat{U}_i = \frac{1}{N}\sum_{k=1}^{N} Y^k\left(x_{k1}^{(1)}, \ldots, x_{k(i-1)}^{(1)}, x_{ki}^{(1)}, x_{k(i+1)}^{(1)}, \ldots, x_{kn_X}^{(1)}\right) \times Y^k\left(x_{k1}^{(2)}, \ldots, x_{k(i-1)}^{(2)}, x_{ki}^{(1)}, x_{k(i+1)}^{(2)}, \ldots, x_{kn_X}^{(2)}\right)$$

Then the n_X first order indices are estimated by

$$\hat{S}_i = \frac{\hat{V}_i}{\hat{\sigma}^2} = \frac{\hat{U}_i - m_{Y^k}^2}{\hat{\sigma}^2}$$

For the second order, the two variables X_i and X_j are not swapped to estimate U_{ij}, and so on for higher orders, assuming that order $< n_X$. Then the $\binom{n_X}{2}$ second order indices are estimated by

$$\hat{S}_{ij} = \frac{\hat{V}_{ij}}{\hat{\sigma}^2} = \frac{\hat{U}_{ij} - m_{Y^k}^2 - \hat{V}_i - \hat{V}_j}{\hat{\sigma}^2}$$

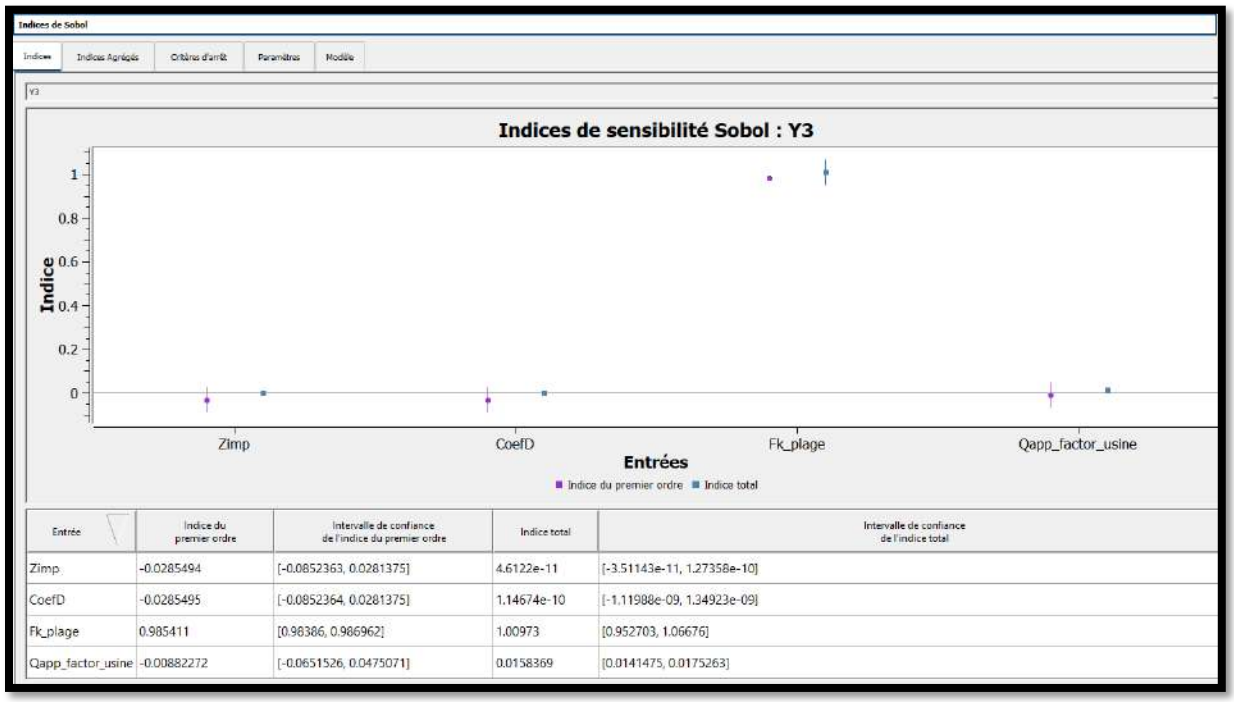
For the n_X total order indices T_i, we only swap the variable X_i between the two samples.

Appendix 31: Sobol indices mathematical construction

Paramètres de l'analyse de sensibilité

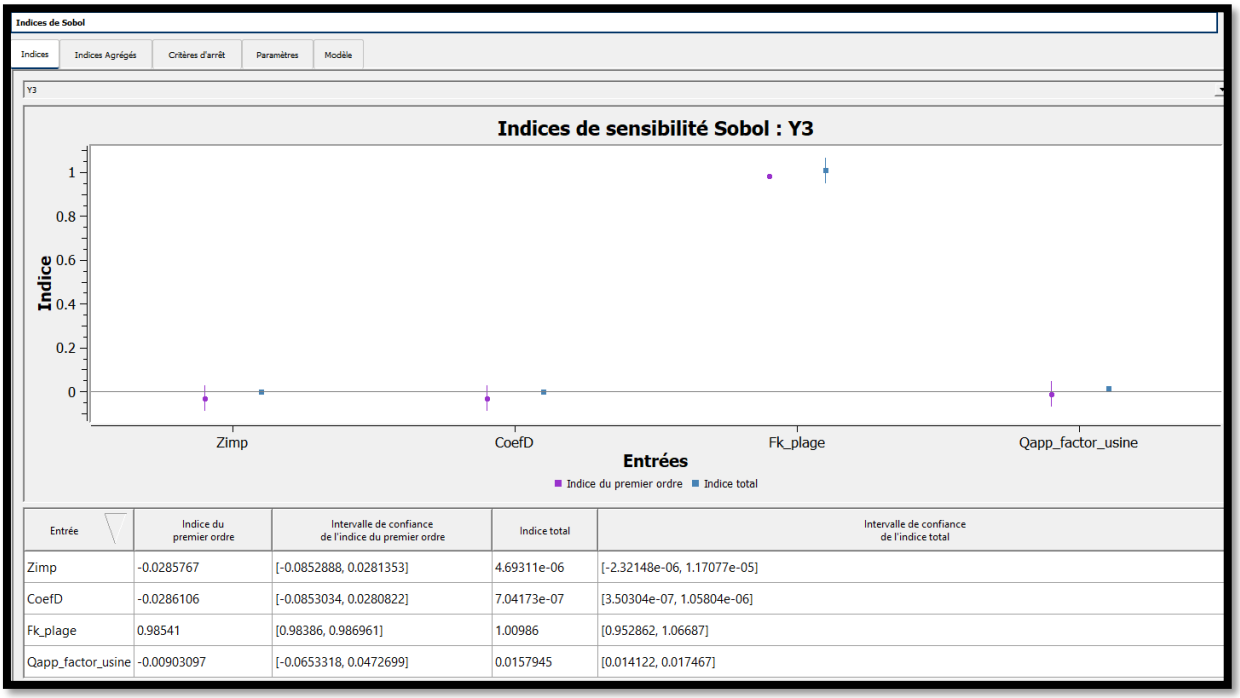
Algorithme	Sobol
Sorties d'intérêt	[Y1,Y2,Y3,Y4,Y5,Y6,Y7]
Taille d'intervalle de confiance maximale	0.01
Durée maximale	- (s)
Nombre d'appels maximum	10000
Taille de réplication	1000
Taille de bloc	1
Graine	0
Niveau de l'intervalle de confiance	0.95

Appendix 32: Sobol indices parameters precision through Persalys GUI



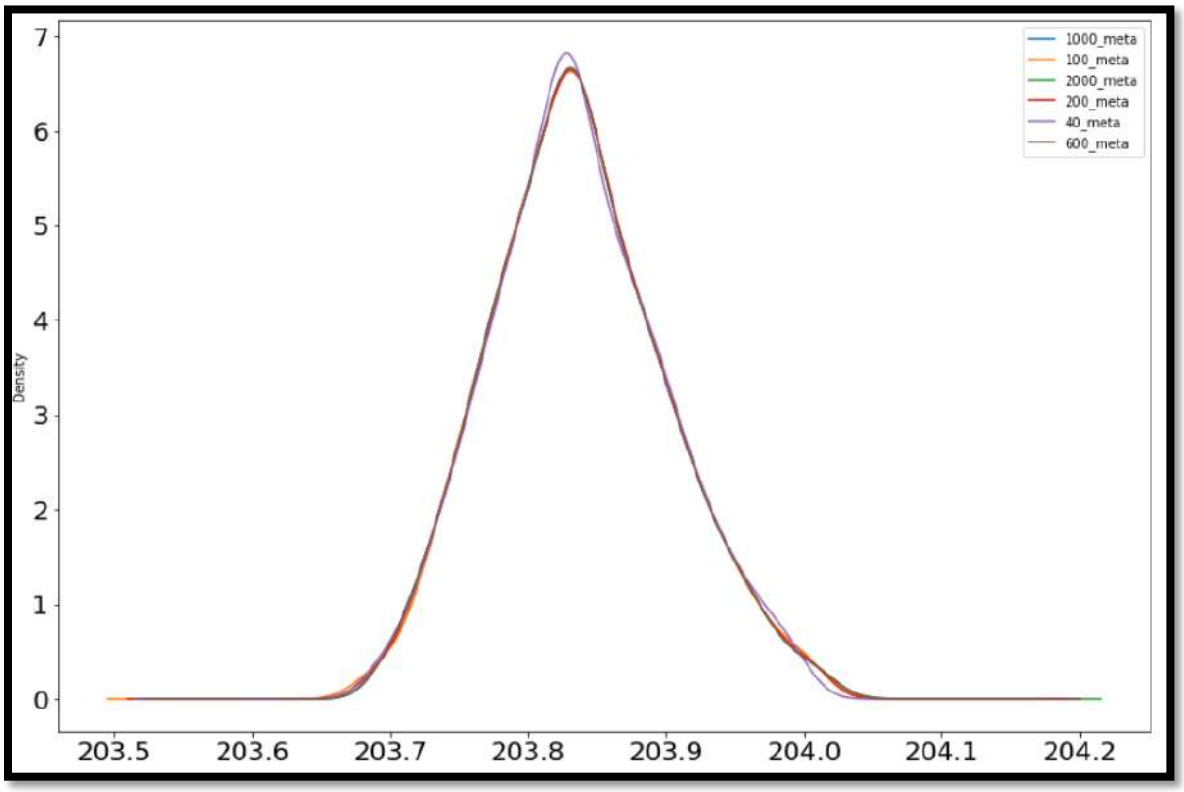
Entrée	Indice du premier ordre	Intervalle de confiance de l'indice du premier ordre	Indice total	Intervalle de confiance de l'indice total
Zimp	-0.0285494	[-0.0852363, 0.0281375]	4.6122e-11	[-3.51143e-11, 1.27358e-10]
CoefD	-0.0285495	[-0.0852364, 0.0281375]	1.14674e-10	[-1.11988e-09, 1.34923e-09]
Fk_plage	0.985411	[0.98386, 0.986962]	1.00973	[0.952703, 1.06676]
Qapp_factor_usine	-0.00882272	[-0.0651526, 0.0475071]	0.0158369	[0.0141475, 0.0175263]

Appendix 33: Sobol indices calculated from the original code



Entrée	Indice du premier ordre	Intervalle de confiance de l'indice du premier ordre	Indice total	Intervalle de confiance de l'indice total
Zimp	-0.0285767	[-0.0852888, 0.0281353]	4.69311e-06	[-2.32148e-06, 1.17077e-05]
CoefD	-0.0286106	[-0.0853034, 0.0280822]	7.04173e-07	[3.50304e-07, 1.05804e-06]
Fk_plage	0.98541	[0.98386, 0.986961]	1.00986	[0.952862, 1.06687]
Qapp_factor_usine	-0.00903097	[-0.0653318, 0.0472699]	0.0157945	[0.014122, 0.017467]

Appendix 34: Sobol indices calculated from the Gaussian process metamodel



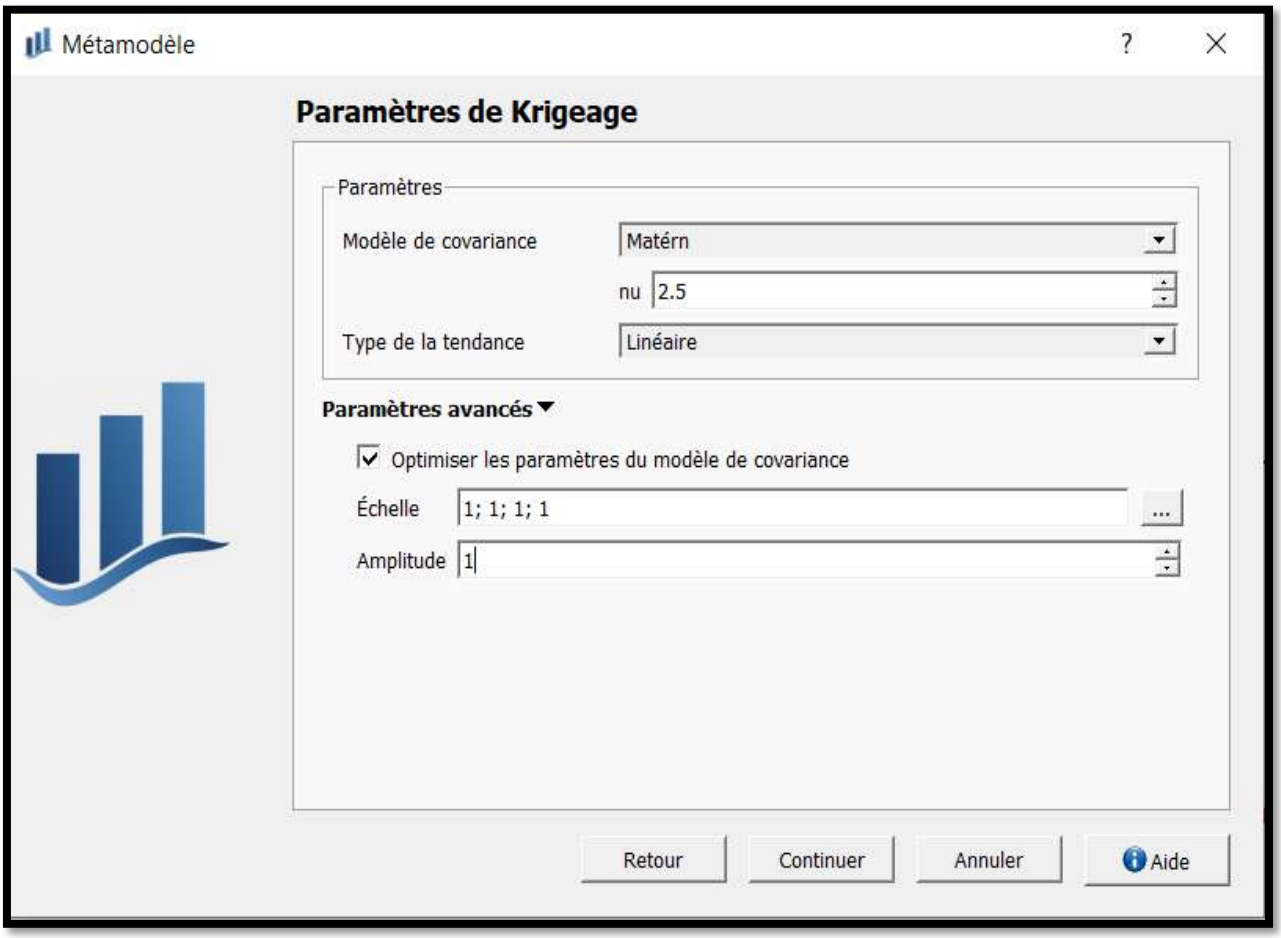
Appendix 35: Response probabilistic distributions of Y3 (P67.200) by the surrogate model (Gaussian process)

Sampling method	Samples	Variable of interest	Minimum value	Maximum value	Mean	Standard deviation	Variability coefficient	Z nominal value	Quantile5%	Quantile 95%
Quasi-Monte Carlo	2000	P67.200	203.686	204.036	203.837	0.063	0.0003	203.841	203.833	203.949

Appendix 36: P67.200 profile output probabilistic configuration in case of n =2000 samples __Crue10

Sampling method	Samples	Variable of interest	Minimum value	Maximum value	Mean	Standard deviation	Variability coefficient	Z nominal value	Quantile5%	Quantile 95%
Quasi-Monte Carlo	2000	P67.200	203.686	204.036	203.837	0.063	0.0003	203.841	203.833	203.949

Appendix 37: P67.200 profile output probabilistic configuration in case of n =2000 samples __Gaussian process



Appendix 38: The Gaussian process configuration and hyperparameters used as a metamodel for the GSA study